



Пловдивски университет „Паисий Хилендарски“
Факултет по математика и информатика
Катедра „Компютърни системи“

**Разработване на интелигентни средства за
работа с виртуализирани културно-
исторически обекти**

АВТОРЕФЕРАТ

на дисертационен труд за присъждане на образователна и научна степен “доктор”
в област на висше образование 4. Природни науки, математика и информатика;
професионално направление 4.6. Информатика и компютърни науки;
докторска програма Информатика

Редовен докторант: Илия Илиев Неделчев
Научен ръководител: проф. д-р Станимир Стоянов

Пловдив, 2025

Дисертационният труд е обсъден и насочен за защита пред научно жури на заседание на катедра „Компютърни системи“ при Факултет по математика и информатика на ПУ „Паисий Хилендарски“, град Пловдив, на 07.11.2025г

Дисертационният труд съдържа 144 страници. Библиографията включва 116 източника. Списъкът на авторските публикации се състои от 2 заглавия.

Защитата на дисертационният труд ще се състои на 06.02.2026г в Заседателната зала на Нова сграда на ПУ „Паисий Хилендарски“, град Пловдив.

Материалите по защитата са на разположение на интересувашите се в секретариата на Факултет по математика и информатика – каб. 330 в Нова сграда на ПУ „Паисий Хилендарски“, всеки работен ден от 08:30 до 17:00 часа.

Научно жури:

Вътрешни членове:

- 1) Проф. д-р Ася Георгиева Стоянова-Дойчева - ФМИ на ПУ „Паисий Хилендарски“
- 2) Доц. д-р Венета Веселинова Табакова-Комсалова – ФМИ на ПУ „Паисий Хилендарски“

Външни членове:

- 3) Проф. д-р Владимир Василев Монов - ИИКТ, БАН
- 4) Проф. д-р Евдокия Николаева Сотирова – Университет „Проф. д-р Асен Златаров“, гр. Бургас
- 5) Доц. д-р Емил Георгиев Делинов - Тракийски университет – Стара Загора

Резервни членове:

- 6) Проф. д-р Сотир Николов Сотиров - Университет „Проф. д-р Асен Златаров“, гр. Бургас
- 7) Доц. д-р Георги Николов Чолаков - ФМИ на ПУ „Паисий Хилендарски“

Автор: Илия Илиев Неделчев

Заглавие: „Разработване на интелигентни средства за работа с виртуализирани културно-исторически обекти“

Пловдив, 2025 год.

УВОД

Дисертацията е посветена на разработването на интелигентни, базирани на знания средства за дигиталното представяне на културно-историческото ни наследство.

Основната цел на дисертацията е изграждането на платформа за дигитализация на културно-историческото ни наследство. Основен компонент на тази платформа ще бъде персонален туристически екскурзовод. За постигане на целта са формулирани следните задачи:

- Анализ на най-новите тенденции за изграждане на такъв тип информационни системи.
- Предложение за усъвършенстване на архитектурата на платформата.
- Разработване на концепция, модел и архитектура на персонализиран туристически екскурзовод.
- Имплементиране на прототип на персонален туристически екскурзовод и провеждане на експерименти с прототипа.

Изследването, проведено по време на докторантурата, преминава през следните етапи:

- Създаване на обща концепция, модел и архитектура на базов туристически екскурзовод.
- Разработване на прототип на базовия туристически екскурзовод, който може да бъде адаптиран за конкретна туристическа дестинация.
- Специфицирани са параметри, отчитащи персонални характеристики на туристите, които могат да се използват за персонализиране на базовия туристически екскурзовод.

- Демонстрира се възможност за адаптиране на базовия туристически екскурзовод за различна приложна област – в случая за мониторинг чистотата на въздуха в района на Пловдив.

В съответствие с целта и задачите е предложена подходяща методика за провеждане на изследването. Дисертацията се състои от увод, четири глави и заключение.

ПЪРВА ГЛАВА: СЪСТОЯНИЕ НА ПРОБЛЕМАТИКАТА

В съответствие с темата на дисертацията следните представляват интерес:

- Специфичните аспекти на съвременните туристи.
- Особеностите на виртуалния туристически гид.
- Интелигентните агенти.

Анализирани са специфичните аспекти на туристическите услуги и туризма. Стратегическите цели в управлението на туристическата дестинация са свързани с подобряване на нейната конкурентоспособност, повишаване удовлетвореността на туристите и постигане баланс в интересите на всички субекти (национална и местна власт, туристически бизнес, местна общност и туристи) [1], както и за постигане на устойчив туризъм, изискващ отговорно поведение на всички страни. Съществено значение за развитието на туризма има качеството на туристическия продукт, който включва различни стоки и услуги за задоволяване потребностите на туристите [2]. Пазарът на туристически услуги е изключително специфичен поради предоставянето на разнообразни по своя вид услуги, включително информационни [3]. Технологиите в туризма са многоизмерен конструкт, като ефективността им се оценява въз основа на четири атрибута: достъпност, информативност, интерактивност и персонализация [4].

Представени са особеностите на виртуалния туристически гид. Основният подход в създаването на виртуалните туристически гидове е изготвянето на система, която да улесни механизма на традиционно търсене на информация и резервиране/закупуване на билети. Създадените системи следва да намалят извършваните дейности от туристите по търсене на информация за туристически атракции, тяхното работно време, подходящ период за посещение и начин за достигане, което ще позволи по-ефективно управление на времето от туристите.

Виртуалният туристически гид е необходимо да има дизайн, който е ориентиран към потребителя, за да се отговори на неговите нужди и да се улесни процеса на използване на приложението. Дизайнът на приложението, който е ориентиран към потребителя, следва да е интерактивен, като се изисква мнението на потребителите по време на разработването му, за да създаде използваем и достъпен продукт. Основният смисъл на виртуалният туристически гид е да се подпомогне туриста да намери необходимата му информация в точното време. Дългото време, необходимо на туристите да се ориентират в непознат град и да установят туристическите атракции и тяхното местоположение, може значително да бъде намалено с използването на виртуален туристически гид.

Анализирана е спецификата на изкуствения интелект и интелигентните агенти. Дефинициите за същността на изкуствения интелект са различни, като той се определя като машинно обучение, което включва програмни подходи, които използват „различни алгоритми и методи, такива като линейна регресия, дървета на решенията, Байесови мрежи, еволюционни алгоритми и изкуствени невронни мрежи“ [5], както и като решение, чрез което се увеличава мощността на компютърните системи. Изкуственият интелект е и способността да се създаде такъв софтуер, чрез който да се „демонстрира мислене, сравнимо с това на хората, като изкуственият интелект е способен да възприема, анализира и взаимодейства със заобикалящата го среда, да се учи от предишен опит и да решава автономно сложни проблеми без намесата на човека“ [6].

Интелигентните агенти по своята същност са независима програма или обект, който взаимодейства със своята среда, като възприема заобикалящата я среда чрез сензори. В изкуствения интелект интелигентните агенти са автономни единици, използващи сензори и задвижващи механизми, за да изследват средата и да изпълняват различни дейности, които са рационални, т.е. максимизират ефективността и постигат най-добрия резултат.

От прегледа на състоянието на проблематиката са направени изводи, които направляваха провеждането на изследването, резултатите от което са представени в дисертационния труд.

Относно спецификите на съвременните туристи:

- Изместване фокуса на туристическото предлагане от традиционни към индивидуални туристически оферти.

- Промяна в предпочитанията на туристите от потребление на унифицирани продукти към използване на индивидуални оферти, за постигане на специфични преживявания.
- Туристите се разграничават на такива, които използват организирани туристически пътувания и индивидуално организиращи посещенията в туристическата дестинация.

Относно виртуалния туристически гид:

- Виртуалният туристически гид следва да предоставя възможност за адаптиране към различни райони.
- С цел проверка работоспособността на резултатите приложената система е адаптирана за гр. Габрово.
- Основната цел за създаването на виртуален туристически гид е решаване на посочените проблеми, свързани с информираността на туристите за различни туристически дестинации (за тестване на гида е избран Габрово като туристическа дестинация).
- Виртуалният туристически гид ще спомогне работата на туристическите информационни центрове (в случая Габрово), поради което е необходимо да бъде насочен към индивидуалния турист и да му предостави възможност за съставяне на индивидуален оптимален маршрут, според личните предпочитания.
- Виртуалният туристически гид трябва да се адаптира при промяна на околната среда, като смяна на метеорологичните условия, отклонение на туриста от зададения маршрут, задръстване по пътя и др.
- Основният сценарий на виртуалният туристически гид е да създаде туристически маршрут, като за целта използва данни от околната среда, като време, местоположение на човек, както и параметри, зададени от туриста – бюджет, разстояние и време с което разполага. При създаването на туристическия маршрут, гидът ще предостави информация на туриста за най-високо оценените туристически обекти.
- Въпреки, че през последните години се установява разработването на все повече информационни системи и приложения в областта на туризма, те са разработени са конкретни туристически услуги, което изисква туристите да използват няколко приложения при посещението на туристическите дестинации. Малко са разработените

приложения, които да предоставят достъп на туристите до всички туристически услуги, да съдържат цялата необходима информация и да използват инструменти за анализиране на ситуацията в туристическата дестинация в реално време.

За успешната реализация на планирания виртуален туристически гид е необходим задълбочен преглед и анализ на съществуващото състояние на интелигентните агенти (Jason, Natural language processing, NLU – Rasa, LLM – OpenAi и Vips).

ВТОРА ГЛАВА: МОДЕЛ И АРХИТЕКТУРА НА ПЕРСОНАЛНИЯ ЕКСКУРЗОВОД

За проверка на предложените модел, архитектура и прототипна реализация на туристически екскурзовод е избран района на град Габрово. Прототипът е адаптиран за обслужване на туристи, посещаващи този район. Габрово има стратегическо местоположение по отношение на достъпност и привличане на туристи, както от Черноморието, така и от Западна България. Общата информираност за Габрово като туристическа дестинация може да се определи като незадоволителна. Основен недостатък е хаотичното представяне на информация за отделните туристически обекти, като места за настаняване, за изхранване, туристически забележителности, възможности за придвижване и др. и ограничена възможност за онлайн комуникация. Мобилни приложения липсват, което е затрудняващ фактор за привличане на туристи. Основната цел за създаването на виртуален туристически гид е решаване на посочените проблеми, свързани с информираността на туристите за туристическата дестинация. Виртуалният туристически гид ще спомогне работата на туристическия информационен център в Габрово, поради което е необходимо да бъде насочен към индивидуалния турист и да му предостави възможност за съставяне на индивидуален оптимален маршрут, според личните предпочитания. В Габрово и околността се съдържат множество туристически обекти и атракции, които поради историческите, кулурните и природните си богатства, следва да бъдат посетени от всеки турист.

Основните функционалности на виртуалния агент са свързани с организирането на оптимален туристически маршрут, който да бъде съобразен със свободното време на туристите, типът на транспорта, с който разполагат, бюджета и разстоянието, което са склонни да изминат. За разлика от традиционните информационни приложения, виртуалният туристически гид не само ще представя списък с всички туристически обекти в Габрово и региона, но ще изготвя персонален туристически план.

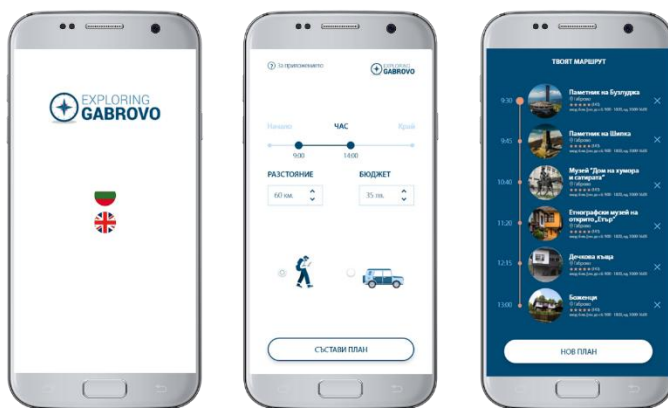
Виртуалният туристически гид е разработено мобилно приложение, което е налично за:

- мобилни устройства с операционна система Android;
- уеб сайт (<https://exploring-gabrovo.com/>).

Системата се състои от няколко части:

- мобилно приложение;
- публичен уеб сайт;
- панел за управление на съдържанието.

Създаденото мобилно приложение е изключително интуитивно, като създава индивидуален план на потребителите, в рамките на две стъпки. На фигура 1 са представени стъпките, които следва да предприеме потребителя на мобилното приложение, за да му бъде изготвен план.



Фигура 1. Стъпки за изготвяне на туристически план

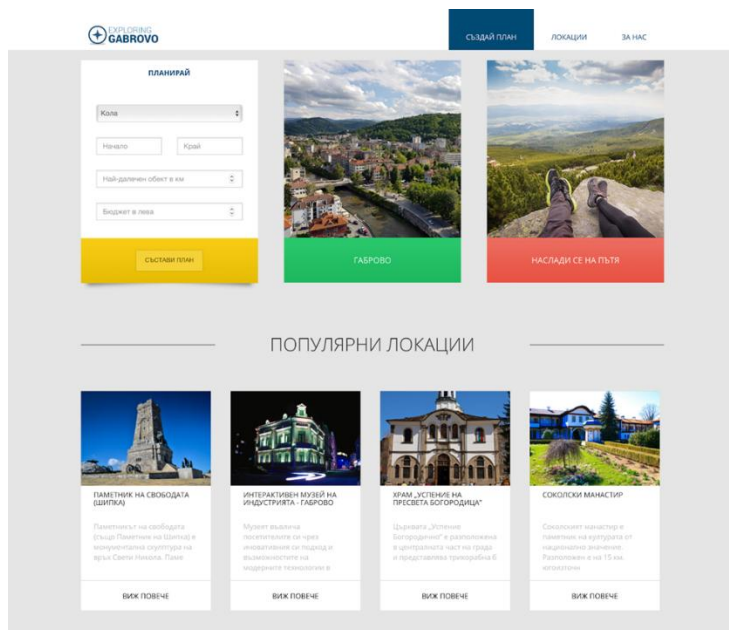
Стъпките за изготвяне на туристически план са:

- Първо, трябва да бъде избран език, защото приложението е планирано, както за български, така и за чуждестранни туристи. За тази цел има опция в интерфейса за избор на български или английски език.
- Второ, потребителят трябва да посочи часовия интервал за своя план, разстоянието, което е склонен да измине, своя бюджет и начин на транспорт.

След натискане на бутона „Състави план“ се появява екран с генериран туристически план. На потребителя се предоставя възможност да изключва обекти, при което да се генерира нов, по-

оптимален план от предходния. Виртуалният туристически гид позволява при маркиране на даден обект от потребителя, да се визуализира допълнителна информация за него, която включва местоположение на обекта, цена за вход, атмосферни условия при които е подходящо посещението, описание на обекта и галерия. При генерирането на туристическия план по зададените параметри от потребителя се взема под внимание и текущото местоположение на потребителя, чрез функцията за геолокация на мобилния телефон. Виртуалният туристически гид отчита и метеорологичната обстановка към момента на генериране на плана, като подбира само туристически обекти, които са подходящи за посещение според спецификата на времето. Изборът на туристически обекти, които са включени във виртуалния туристически гид, се базира на получената за тях оценка от google ratings.

Публичният уеб сайт има същата функционалност като мобилното приложение, като включва също списък с всички туристически обекти, които са на разположение в системата.



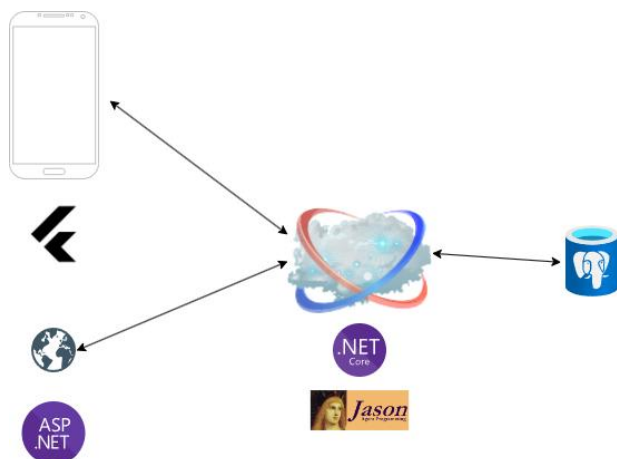
Фигура 2. Начална страница на публичния уебсайт

На фиг. 2 е представена началната страница на публичния уебсайт, на която се съдържа и формата, чрез която потребителите могат да генерираат туристически маршрут. На началната страница е представен и списък с популярни локации. В публичния уебсайт се съдържа списък с всички туристически обекти в Габрово и областта. Обектите в списъка са сортирани по оценка на обекта, като се започва от най-високо оценените обекти.

Панелът за управление на обектите е предназначен единствено за потребители, които имат права да редактират и добавят съдържание. Такива потребители могат да бъдат администратори на системата, служители на туристически информационен център или служители на конкретен туристически обект. В панела за управление на съдържанието може да се добавят нови обекти, да се редактира съдържание на вече добавени обекти, както и да се изтриват съществуващи такива. В този панел могат да се добавят и премахват потребители от администраторите, като всеки потребител има или администраторска или редакторска роля. Екранът на панела за управление на обектите включва списък на всички обекти, като до всеки обект има бутон за редакция и бутон за изтриване. На същата страница има и бутон за добавяне на нов обект. При натискане на добавяне или редакция се отваря прозорец за попълване или редакция на информация за обекта.

На фиг. 3 е представена архитектурата на туристическия екскурзовод. Мобилното приложение прави заявки към уеб ресурс, за да вземе необходимата информация за обектите и да генерира туристически маршрут на потребителя. Уеб приложението прави заявки към база данни, в която се съхранява информация за обектите. Мобилното приложение е реализирано чрез използване на технологията Flutter, която е комплект за разработка на софтуер за потребителски интерфейс с отворен код, създаден от Google [7]. За разработване са използвани и приложения за Android, iOS, Linux, Mac, Windows, Google Fuchsia и мрежата от една кодова база.

Уеб приложението е реализирано чрез използване на технологията ASP.NET Core – безплатна уеб платформа с отворен код и наследник на ASP.NET, разработена от Microsoft [8]. Уеб приложението се състои от сайт и REST API, като REST API обработва заявките изпратени от мобилното приложение [9].

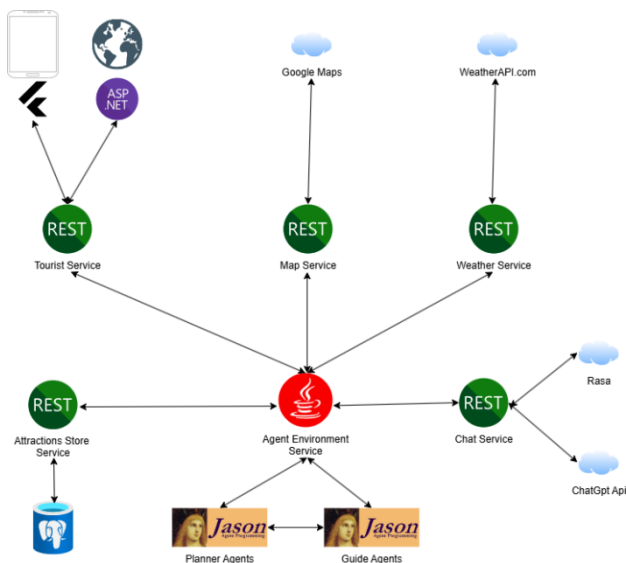


Фигура 3. Визуализация на архитектурата на приложението

За хостинг на уеб приложението е използвана облачната платформа Heroku, а за инструмент за виртуализация се използва Docker [10]. Docker е набор от платформи като сервизни продукти, които използват виртуализация на ниво OS, за да доставят софтуер в пакети, наречени контейнери.

За база данни се използва релационната база данни с отворен код PostgreSQL, която поддържа множество типове данни, транзакции, криптиране, регулярни изрази и др. и се използва широко, включително от големи компании [11].

Архитектурата на виртуалния туристически гид съдържа интелигентни агенти, като решението е взето поради факта, че една от идеите, която стои при създаването на интелигентните агенти е те не просто да следват определен алгоритъм, както е при „типичните“ компютърни програми, а да действат, както би постъпил човек. Основните свойства на интелигентните агенти са автономност, проактивност, реактивност и социални умения. Виртуалният туристически гид е многопластова система [12], която включва мултиагентна среда реализирана на framework Jason. Agent Environment Service е Java околната среда за Jason агентите, тя е включва в себе си Rest Api реализиран чрез Javalin. Това позволява на агентите да комуникират с външният свят.



Фигура 4. Мултиагентна среда на виртуалния туристически ГИД

Използвани са два вида Jason агенти:

- **Planner Agents** – Ролята на която е да управлява Guide агентите, когато дойде заявка за нов план Planner Agents създава Guide агент. Също така се грижи за освобождаването на Guide Agent когато не е нужен.
- **Guide Agent** – това агент, който се грижи за индивидуалния план на туристът. За всеки турист има отделен Guide agent, този агент генерира план, следи промените на околната среда, локацията, метеорологичните особености, работното време и на база на това при нужда сменя плана. Неговата основна цел е да заведе потребителя през максимално оценени атракции до края на времето с което потребителя разполага. Този агент реагира проактивно на промените на средата.

За архитектура на агентите е избрана BDI – базирана архитектура (Belief-desire-intention). Посоченият избор е направен поради факта, че този тип архитектура се приближава максимално до човешкото поведение при вземане на решение.

Потребителска част – първоначално мобилното приложение за потребителите е изработено с Xamarin – позволява изработка на платформено независими приложения, което означава, че след изработка ще има поддръжка както за Android, така и за iOS. Xamarin е собственост на Microsoft и поддържан от Microsoft, това е една от причината за избора на тази технология. След тестването на Xamarin за мобилната технология е избрано Flutter. За изработка на уеб сайта е използван ASP.NET Core MVC приложение с Blazor. Blazor се компилира до Web Assembly, което се изпълнява на уеб браузъра на клиента. Мобилното приложение и уеб сайта ще споделят голяма част от логиката, която ще бъде написана на програмен език C#. И двете приложения ще се обръщат към Уеб приложение, което ще съхранява данните за потребителя и обектите. Уеб сайтът ще съдържа и допълнителна част, която ще служи за администрация на потребители и обектите от потребители с роля администратор.

Сървърна част – сървърната част се състои от уеб приложение, база данни и Java REST приложение с Jason среда [13]. Уеб приложението използва ASP.NET Core технология за реализация на REST приложение. Това приложение ще съхранява и администрира данните за потребителите и обектите и ще прави заявки към Jason средата за изготвяне на индивидуалните туристически планове. За база данни се използва PostgreSQL. PostgreSQL е релационна бази данни с отворен код. Jason средата ще е интегрирана с библиотека Spark, чрез която ще се реализира REST приложението.

Освен мултиагентната среда има още 4 уеб приложения и едно мобилно приложение. Attractions Store е реализирано чрез използване на технология ASP:

- NET Core е приложение, което съдържа информацията за туристическите обекти. То има връзка към Sql Server база данни, в която се съдържа информация за туристическите обекти и техните характеристики, като работно време, галерия от снимки, дали са подходящи за дъждовно време, дали са на открито или закрито, gps локация.
- Tourist Service също е реализиран на технология ASP.NET Core. Целта на това уеб приложение е да приема заявките от туристите направени през мобилното приложение, както и регулярна информация за промените при туриста (като текущо местоположение и промяна в изискванията за съставяне на плана).

- Map Service се грижи за съхраняване на разстоянията между обектите, като може да връща информация дали потребителя не е близо до даден обект. Приложението осъществява връзката и с Google Routes сървиса, с която помощ се изчислява колко време ще отнеме на потребителя да достигне до даден обект на база дали е с кола или пеша.
- Weather Service връща информация за метеорологичните особености при даден обект и където се намира туриста. Този сървис прави обращения към WeatherApi.Com на определен интервал и кешира информацията, като по този начин намалява извикванията към WeatherApi.Com.
- Chat Service управлява чата интеракцията с потребителя, като потребителят винаги може да зададе въпрос в чат прозорец и да получи отговор. Съществуват два вида чат. Rasa са предефинирани сценарии, като Rasa Nlu (Natural Language understanding), за да се разбере намерението на потребителя. Ако Rasa сценарият не може да обработи дадената потребителска заявка, то чатът се препраща към ChatGp API, които използва LLM за да обработи по генерална заявка. Когато потребителят доближи даден обект, то неговите GPS координати са в близост на координатите на обекта и се задейства чат за съответният обект, като отново се задейства първо Rasa сценарии за съответният обект, ако разговора обаче излезе от дефинираният сценарии се прави обръщение към Chat GPT.

Мобилното приложение е реализирано чрез използването на технологията flutter, като постоянно изпраща координатите на потребителя след стартиране на изпълнението на плана. Така при нужда и ако потребителя се отклони, Guide агента праща съобщение до Tourist Service, съответно той уведомява потребителя. Също така му предлага на промяна на плана при промяна на посоката, метеорологичните условия или когато даден туристически обект затваря за деня.

При обработка на заявка за план се включват следните дейности:

- Заявка от мобилно приложение или уеб сайт за изготвяне на план. Изпраща се идентификатор на потребителя, локация, време за започване на плана и време за приключване на плана.
- Прави се заявка към базата данни за взимане на профила на потребител по идентификатор.

- Отговор на заявката към базата данни за потребителският профил.
- Прави се заявка към Jason средата за изготвяне на туристически план за потребителя с потребителският профил, локация, време за започване на план и време за приключване на план.
- Отговор, че плана е приет успешно и е поставен в опашката на агента за обработка.
- Заявка от Jason средата към Уеб приложението за информация за обекти в близост до локация и работно време в интервала на плана.
- Заявка към базата данни за информация за обекти, които са близост до локацията и имат работно време в оказаният интервал.
- Отговор от база данни към уеб приложението с информация за обектите.
- Отговор от уеб приложението към Jason средата с информация за обектите.
- Изпращане на изготвеният план от Jason средата към уеб приложението.
- Отговор от уеб приложението, че е получило плана успешно.
- Отговор от уеб приложението към потребителя включващ изготвеният план.

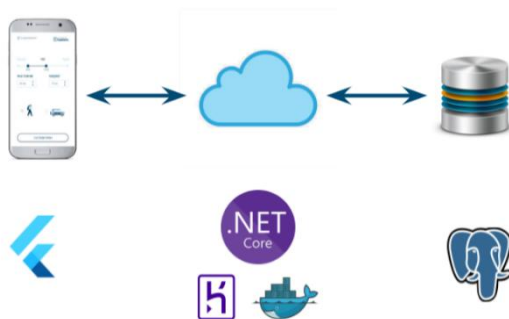
Процесите в системата на виртуалния туристически гид включват:

- Процес преди потребителя да се регистрира или автентикира в системата. Това е процеса, при който потребителят влиза в системата, но не чрез своя профил (или все още няма такъв). Той ще вижда формата за регистрация и вход, ще може да разглежда различните обекти, но няма да има достъп до модула за изготвяне на маршрут.
- Процес на регистрация и попълване на профил на нов потребител. При този процес, потребителят ще създаде своя профил и ще попълни своите предпочитания, чрез въпросник.
- Процес на изготвяне на маршрут на влязъл в система потребител. Този процес може да бъде стартиран веднъж щом потребителят влезе в своя профил и на база на предоставените данни от него да се изготви персонален маршрут за него.
- Процес на администриране на съдържание в системата. Този процес може да се извърши само от потребител с роля администратор.

ТРЕТА ГЛАВА: ПРОТОТИПНА РЕАЛИЗАЦИЯ

За програмната реализация на прототипа е използван Jason – инструменти за изграждане на мултиагентна система. Jason е интерпретатор с отворен код, който разширява AgentSpeak, позволяващ да се изгради и среда за агентите, чрез програмния език Java. А самите агенти се описват чрез Prolog. Напредъкът в областта на изкуствения интелект ще изисква разработване на агенти, които безпроблемно съчетават реактивно и проактивно поведение. Тези агенти не само ще отговорят на непосредствените нужди, но и ще предвидят и подготвят бъдещи предизвикателства [14]. Посочените свойства на интелигентните агенти позволяват на туристическия гид ефективно сътрудничество между отделните агенти. Постигането на целите на туристическия гид също се основава в мултиагентна среда. Важно е да се уточни, че използваните агенти могат да се адаптират при промяна в състоянието на различните туристически обекти, които са използвани.

Използваната архитектура на туристическия гид по отношение на агентите е BDI - Belief-desire-intention (Вярване – Желание – Намерение). Посочената архитектура е основна тема в множество проведени изследвания, свързани с интелигентните агенти. В BDI архитектура вярванията представляват информацията на агента за неговата среда, други агенти и себе си, целите (желанията) са състояния за постигане, а намеренията са ангажменти за постигане на конкретни цели [83]. Архитектурата на системата в прототипа е представена на фигура 5.

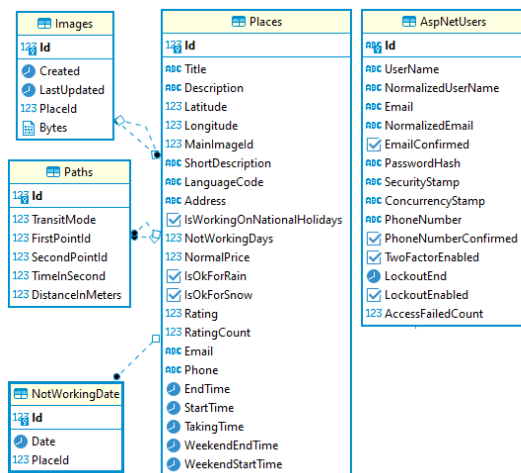


Фигура 5. Архитектура на системата на прототипа

Видно от представената архитектура на фигура 5, мобилното приложение прави заявки към уеб ресурс, за да вземе информация за обектите и за да се генерира туристически маршрут за клиента. От своя страна уеб приложението прави заявки към база данни, в която се съхранява информацията за обектите. Мобилното приложение е реализирано с използване на Flutter (софтуер за потребителски интерфейс с отворен код, създаден от Google, който се използва за разработване на приложения за Android, iOS, Linux, Mac, Windows, Google Fuchsia и мрежата от една кодова база).

Уеб приложението е реализирано с използване на технологията ASP.NET Core – безплатна уеб платформа с отворен код и наследник на ASP.NET, разработена от Microsoft. Уеб приложението се състои от сайт и REST API, като REST API обработва заявките изпратени от мобилното приложение. За хостинг на уеб приложението се използва облачната платформа Heroku, а за инструмент за виртуализация се използва Docker (набор от платформи като сервизни продукти, които използват виртуализация на ниво OS, за да доставят софтуер в пакети, наречени контейнери). За база данни се използва Post-greSQL (релационна база данни с отворен код, поддържаща множество типове данни, транзакции, криптиране, регулярни изрази и др.).

На фигура 6 е представена диаграма на схемата на базата данни.



Фигура 6. Диаграма на схемата на базата данни

Използвани са следните таблици:

- В таблицата Images се пазят снимките към туристическите обекти (съхраняват се като байтови масиви в базата).
- Таблицата Paths съдържа разстоянието между два туристически обекта за даден вид транспорт.
- В таблицата NotWork-ingDate се съхраняват датите, през които обектът не работи.
- Таблицата Places съхранява информацията за туристическите обекти. ТаблицатаAspNetUsers пази информация за потребителите.

За изработка на мобилното приложение на клиентската част е използван Flutter – позволяващ изработка на платформено независимо приложение, което означава, че след изработката има поддръжка за Android за iOS. Основна причина за избора на Flutter е, че е собственост на Google е поддържан от компанията. За изработка на уебсайта от клиентската част е използван ASP.NET Core MVC приложение с Blazor. Blazor се компилира до Web Assembly, което се изпълнява на уеб браузър на клиента. Мобилното приложение и уеб сайта ще споделят голяма част от логиката, която ще бъде написана на програмен език C#. Посочените две приложения се обръщат към уеб приложение, което съхранява данните за потребителя и туристическите обекти. Уеб сайтът съдържа и допълнителна част, имаща за цел да позволи администрация на потребители и обекти от потребителя, с роля на администратор.

Сървърната част на туристическия гид се състои от уеб приложение, база данни и Java REST приложение с Jason среда. Уеб приложението използва ASP.NET Core технология за реализация на REST приложение, което съхранява и администрира данните за потребителите и обектите и прави заявки към Jason средата за изготвяне на индивидуалните туристически планове. За база данни се използва PostgreSQL. – релационна бази данни с отворен код. Jason средата е интегрирана с библиотека Spark, чрез която се реализира REST приложението. Компонентите на туристическия ги са:

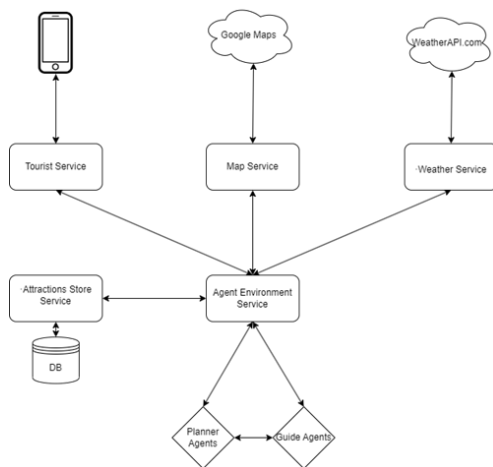
- Агенти за планиране – когато нов турист започне да използва мобилното приложение, му се назначава агент за планиране. Основната задача на агента за планиране е да създаде агент-екскурзовод, специално пригоден за нуждите на туриста.
- Агенти-екскурзоводи – отговарят за всички аспекти на планирането на пътуването, включително събиране на информация, генериране на планове, наблюдение на времето и

местоположението на потребителя и коригиране на плановете, ако е необходимо. Те използват Jason Framework, за да управляват своите убеждения, желания и намерения, което им позволява да адаптират плановете си въз основа на различни фактори.

- Събиране на информация – агентът-екскурзовод събира всички необходими данни за планиране на пътуването, използвайки различни RESTful API услуги. Това включва данни, свързани с туристически атракции и метеорологични условия. Агентът използва тези данни, за да взема информирани решения и да предоставя подходящи препоръки.
- Мобилен приложение – служи като основен интерфейс за взаимодействие на потребителя с агента-екскурзовод. Чрез мобилното приложение потребителите могат да въвеждат своите предпочитания, като бюджет, времева рамка и ограничения за разстояние. Те могат също да преглеждат генерирания план за пътуване, да предоставят обратна връзка и да изискват промени.
- Генериране на план – въз основа на предпочитанията на потребителя и данните в реално време, събрани от различни API услуги, агентът-екскурзовод създава първоначален план за пътуване, който отчита ограниченията и факторите на потребителя, като туристически атракции, настаняване, хранене и транспорт.
- Мониторинг на времето и проследяване на местоположението на потребителя – агентът-екскурзовод непрекъснато следи метеорологичните условия и местоположението на потребителя, използвайки API услуги и GPS данни от устройството на потребителя. Това позволява на агента да коригира плана за пътуване в реално време въз основа на променящите се метеорологични условия или движенията на потребителя.
- Корекция на плана – агентът-екскурзовод може да коригира плана за пътуване въз основа на обратна връзка, заявки или промени в данните в реално време от потребителя. Например, ако потребителят реши да удължи престоя си на определено място, агентът може да промени плана, за да се съобрази с тази промяна, като същевременно вземе предвид предпочитанията и ограниченията на потребителя.

В посочената система агентът-планьор се фокусира върху създаването на агенти-екскурзоводи, докато агентите-екскурзоводи се справят с всички аспекти на планирането на пътуването и комуникацията с потребителя чрез мобилното приложение. Комбинацията от Jason Framework MAS и RESTful API услуги позволява разработването на високо адаптивно и персонализирано приложение за туристически пътеводител, което може да реагира на променящите се условия и потребителски вход, осигурявайки оптимално пътуване, съобразено с уникалните нужди и предпочитания на всеки потребител.

Агентът за планиране е интелигентен софтуерен обект, който координира и управлява задачи, като създава, наблюдава и контролира други агенти или процеси. Използвайки възможности за разсъждение и вземане на решения, агентите за планиране обработват входящи заявки, разпределят ресурси и определят цели за подчинените агенти. В многоагентните системи агентите за планиране играят ключова роля в управлението на сложни задачи и постигането на системни цели, като динамично се адаптират към промените в средата и съответно коригират поведението на отделните агенти.



Фигура 7. Архитектурата на услугите и агентите

Чрез ефективно оркестриране на задачи и ресурси, агентите за планиране могат да подобрят цялостната производителност и устойчивост на системата. Сегмент от програмната реализация на агента за планиране е даден на Фиг. 8.

```
+tourist(ID,X,Y,W,B,T):true  
<- .print("нов турист");  
.create_agent(ID, "guide.asl");  
.send(ID tell init(ID));  
.send(ID tell gps(X,Y));  
.send(ID tell weather(W));  
.send(ID tell budget(B));  
.send(ID tell time(T));  
.send(ID tell startGuide(ID)).  
+finishPlane(X):true  
<- .print("Финиш на гида");  
.kill_agent(X).
```

Фигура 8. Сегмент от програмата реализация на агента

Гид агентът е интелигентен софтуерен обект, предназначен да помага на потребители, като например туристи, като предоставя персонализирани препоръки и планове въз основа на техните предпочитания, ограничения и фактори на околната среда. Използвайки усъвършенствани възможности за разсъждение и вземане на решения, гид агентите се адаптират към динамична среда и приспособяват своите насоки, за да създадат задоволително изживяване за потребителите. В многоагентните системи гид агентите взаимодействат с други агенти, като например агенти за планиране, за да координират и изпълняват ефективно задачи. Чрез използване на своята база знания и адаптиране към промените в реално време, гид агентите предоставят високо ниво на персонализиране, подобрявайки потребителското изживяване и предоставяйки ценна помощ в различни области. На фиг. 9 е представен фрагмент от код, описващ агент-екскурзовод в Jason. Този фрагмент от код описва агент-екскурзовод в Jason, който помага на туристите, като създава персонализирани планове въз основа на предоставената информация.

```

@init[atomic]
+init(ID): true
<- init(ID).
// Ново местоположение.
+gps(X,Y): true
<- gps(X,Y).
+weather(W): true
<- weather(W).
+budget(B): true
<- budget(B).
+time(T): true
<- time(T).
+startGuide(ID): true
<- startGuide(ID);
!createPlan.
// Завършване на туристическия план или време.
+finish(X): true
<- send(planer,tell,finishPlane(X)).
+visit(A): true
<- visit(A).
+checkTime(T):true
<- adjustPlanAccordingTime(T).
+changeDirection(X,Y):true
<- adjustPlanAccordingDirection(X,Y).
+changeWeather(W):true
<- adjustPlanAccordingWeather(W).
+!createPlan:true
<- createPlan.

```

Фигура 9. Фрагмент от код, който описва агент-екскурзовод в Jason

Мултиагентният туристически пътеводител, подсилен от възможностите на Rasa, Jason, ChatGPT и .NET 6 REST API, е доказателство за симбиотичната връзка между технологиите и туризма, като той не само разпространява информация, но и насърчава ангажираността, предоставяйки не само упътвания, но и културното опознаване на гр. Габрово. В последната версия на приложението остава разработването на мобилно приложение като многофункционална система за туристически справочници, предназначена да предложи на пътешествениците цялостно изживяване при опознаването на Габрово. Приложението предлага широк набор от функции, включително навигация, информация и местни ресурси, но интеграцията на чатбот и

гласов бот подобрява неговата полезност. Приложението служи като универсален инструмент, отговарящ на множество нужди на пътешествениците. То обхваща функционалности като помощ при навигация, планиране на събития, препоръки за местна кухня и актуализации в реално време за атракции и събития. Потребителите могат да получат достъп до богата информация за историята, културата и традициите на Габрово чрез интуитивен и лесен за ползване интерфейс.

В последната версия на приложението като ценни допълнения към мобилното приложение, са включени чатботът и гласовият бот, предлагащи разговорен слой, който безпроблемно се интегрира с по-широките функционалности на приложението. Тези агенти, управлявани от изкуствен интелект, действат като отзивчиви и информативни спътници, обогатявайки преживяването на пътешественика с персонализирани взаимодействия. Когато потребителите достъпват специфични менюта в приложението, то автоматично отваря режим на разговор. Този режим предоставя набор от функции и информация за Габрово и неговите атракции. След това пътуващите могат да инициират запитвания, да търсят препоръки или да участват в непринуден диалог, за да подобрят разбирането си за региона.

Системата използва две усъвършенствани технологии за разговорен изкуствен интелект, Rasa и ChatGPT, по оркестриран начин. Когато потребител инициира разговор или запитване, системата първо се консултира с Rasa за разпознаване на намеренията и управление на диалога. Ако Rasa определи, че запитването на потребителя не може да бъде адекватно отговорено, тя грациозно преминава към резервна стратегия, като извиква ChatGPT API. Rasa Open Source е платформа с отворен код за разговори с изкуствен интелект, която ви позволява да разбирате и провеждате разговори, както и да се свързвате с канали за съобщения и системи на трети страни чрез набор от API. Тя предоставя градивните елементи за създаване на виртуални (цифрови) асистиенти или чатботове. Обучението на Rasa модели за осъществяване на интелигентни разговори е ключов аспект на мултиагентния туристически пътеводител. За да постигнем това, използваме стандартния Rasa REST API за обучение и извод, но нашата уникална архитектура включва специално разработен Flask REST API за улесняване на комуникацията с Rasa агенти, всеки от които е зареден с отделни модели, пригодени за различни цели. Първоначалните дейности са свързани с обучението на Rasa модели, критична стъпка за

гарантиране, че чатботът и гласовия бот са добре подготвени да ангажират ефективно потребителите. Универсалната рамка на Rasa позволява да се настройват фино моделите за прецизно разпознаване на намерения, извличане на обекти и управление на диалог. Този обширен процес на обучение дава възможност на агентите да разбират широк спектър от потребителски входни данни, вариращи от общи запитвания за Габрово до силно специфични въпроси, свързани с отделни атракции.

Същността на архитектура се основава на персонализирания Flask REST API, щателно разработен, за да организира комуникацията с Rasa агенти. Този API служи като централен хъб, чрез който потребителските заявки, запитвания и взаимодействия се насочват към съответния Rasa модел, осигурявайки контекстуално релевантен и точен отговор. Освен това, Flask REST API безпроблемно комуникира с .NET 6 REST API, което допълнително подобрява възможностите на системата.

Използваният е код, дефиниращ уеб приложение на Flask, което служи като комуникационен мост между персонализирано приложение и агентите на Rasa chatbot (фиг. 10). Този код дефинира уеб приложение на Flask, което служи като комуникационен мост между вашето персонализирано приложение и агентите на Rasa chatbot. Той инициализира и управлява Rasa агенти за различни сценарии и ви позволява да изпращате съобщения до тези агенти за генериране на отговори. Кодът предоставя RESTful API, използващ Flask за инициализиране и взаимодействие с агенти на Rasa чатбот за различни сценарии. Той ви позволява да изпращате съобщения до тези агенти и да извличате техните отговори, което го прави ключов компонент на вашата система за многоагентни туристически пътеводители. При многоагентния туристически пътеводител е използван SQL Server като надеждно решение за съхранение на Rasa модели. Структурата на таблиците, приета за тази цел, се състои от два основни елемента: заглавия на модели и съответните физически пътища на диска. Този структуриран подход към съхранението на модели повишава ефективността, мащабируемостта и надеждността, допринасяйки за подобро потребителско изживяване. По същество този код представлява ключовото взаимодействие между нашето приложение за туристически пътеводител и чатбота, базиран на OpenAI GPT. Той позволява на потребителите безпроблемно да търсят информация, да задават въпроси и да участват в смислени диалози за Габрово и неговите забележителности. Това взаимодействие е ключов елемент от нашия мултиагентен туристически пътеводител, обогатявайки

потребителското изживяване, като предоставя информативни и контекстуално релевантни отговори.

```
from flask import Flask, jsonify, request
import rasa.core.agent
from waitress import serve
def create_app(test_config=None):
    app = Flask(__name__)
    agents = dict()
    endpoints = "endpoints.yml"
    @app.route('/initialize/<int:scenario_id>', methods=['POST'])
    async def initialize(scenario_id):
        model_path = request.get_json()['modelPath']
        endpoint_config = rasa.core.utils.read_endpoints_from_path(endpoints)
        # Умишлено презаписване на настройката на модела в конфигурацията
        на крайната точка засега
        endpoint_config.model = None
        agents[scenario_id] = await rasa.core.agent.load_agent(
            model_path = model_path,
            endpoints = endpoint_config)
        return "", 204
    @app.route('/send-message/<string:conversation_id>/<int:scenario_id>',
        methods=['POST'])
    async def send_message(conversation_id, scenario_id):
        message = request.get_json()['message']
        if scenario_id not in agents.keys():
            return "", 403
        results = []
        responses = await agents[scenario_id].handle_text(message, None,
            conversation_id)
        for response in responses:
            if "text" in response:
                results.append(response["text"])
        return jsonify(results)
    return app
serve(create_app(), host="rasa.local", port=8080)
```

Фигура 9. Фрагмент от код, който описва агент-екскурзовод в Jason

В обобщение следва да се посочи, че използваният подход интегрира най-съвременни технологии, с включването на Rasa за интелигентна функционалност на чатбот и интегрирането на ChatGPT за динамични и контекстно-осъзнати разговори. Основата на системата е Jason, стабилна агентно-базирана рамка, която играе ключова роля в организирането на взаимодействията и предоставянето на високо персонализирани преживявания на потребителите.

Забележителни аспекти на създадената система включват внедряването на .NET 6 REST API за улесняване на безпроблемната комуникация, позволявайки на Jason да взаимодейства с Rasa и ChatGPT. Подчертана е и критичната роля на SQL Server за ефективното управление на заглавията на моделите и физическите местоположения, което допринася значително за бързината и надеждността на нашия туристически пътеводител.

Системата се придържа към проактивен подход, като инициира разговори въз основа на GPS местоположението на потребителя, като по този начин гарантира, че пътуващите получават навременна и контекстуално релевантна информация за близките атракции. Освен това, ние безпроблемно интегрирахме мобилно приложение, обогатено с гласови и чатбот функционалности, подобрявайки ангажираността и достъпността на потребителите.

По време на процеса на разработка изяснихме практическото приложение на Rasa и ChatGPT, демонстрирайки ключовите им роли в улесняването на разговорите, отговарянето на запитвания и предоставянето на подробна информация за Габрово и неговите забележителности.

Нашият мултиагентен туристически пътеводител представлява сливане на авангардни технологии и прагматична полезност, замислен да предложи на пътуващите интелигентен и завладяващ спътник за опознаване на Габрово и околните региони. Чрез използване на потенциала на LLM, NLU и агентно-базираните рамки, ние създадохме система, умела да отговори на разнообразните изисквания на туристите, предоставяйки персонализирани и богати на информация преживявания.

ЧЕТВЪРТА ГЛАВА: ПРОТОТИП НА МУЛТИАГЕНТНА ПЛАТФОРМА ЗА ПОДПОМАГАНЕ ОБЕКТИВНИЯ АНАЛИЗ НА КАЧЕСТВОТО НА ВЪЗДУХА

В главата се демонстрира възможностите на разработените архитектура и прототип на персонален екскурзовод да се адаптира за различна приложна област – мониторинг на въздуха в района на Пловдив.

Предложеният прототип се основава на факта, че поради различни причини – като географско местоположение, климат и климатични промени, това, че е център на зеленчукопроизводството в България, както и наличието на голям медно-цинков завод и депа за отпадъци в близкия район – Пловдив е сред най-замърсените региони в България. Опазването на качеството на въздуха в Пловдив се превръща във все по-належащ проблем. Тъй като мониторингът на качеството на въздуха е чувствителна тема за Пловдив, често има противоречиви измервания, предположения, мнения и публикации относно надеждността на данните и разположението на инструментите за мониторинг. С разработването на прототипа се цели да се създаде мултиагентна платформа, която, като се има предвид този контекст, ще подпомогне обективен анализ на качеството на въздуха чрез:

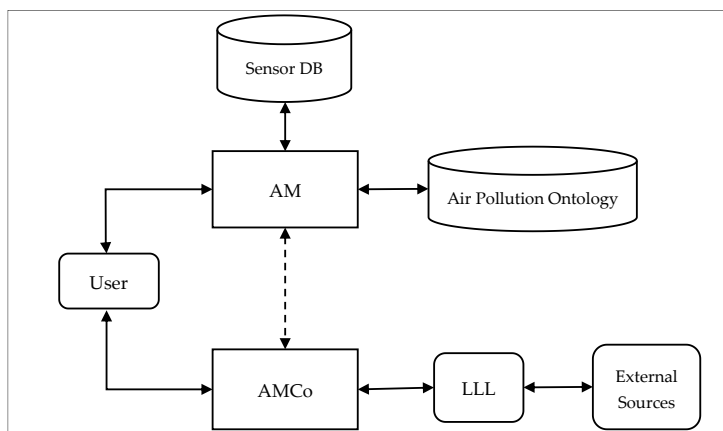
- Докладване на нашите измервания.
- Включване на допълнителна информация от външни източници.
- Сравняване на данни и анализ на отклонения и несъответствия.

Платформата е хибридна, тъй като интегрира методи както от символен, така и от субсимволен изкуствен интелект. Ядрото ѝ се състои от два лични асистента: единият, базиран на символни методи на изкуствен интелект, а другият - на субсимволични методи на изкуствен интелект. В еволюцията на изкуствения интелект тези два основни подхода често са били разглеждани като конкурентни и взаимно изключващи се. Концепцията за интегриран изкуствен интелект обаче напоследък набира скорост, подчертавайки ползите от комбинирането на методи от двете парадигми. За целите на разработване на прототипа смятаме, че хибридната платформа е подходящо решение.

Първият асистент, предназначен да обработва нашите измервания и да работи с нашата база знания, е разработен като BDI-агент,

използващ технологията JaCaMo. Вторият асистент, натоварен със задачата да включва външни източници и да извършва анализи, е реализиран като RaAct агент, използващ технологията LangChain.

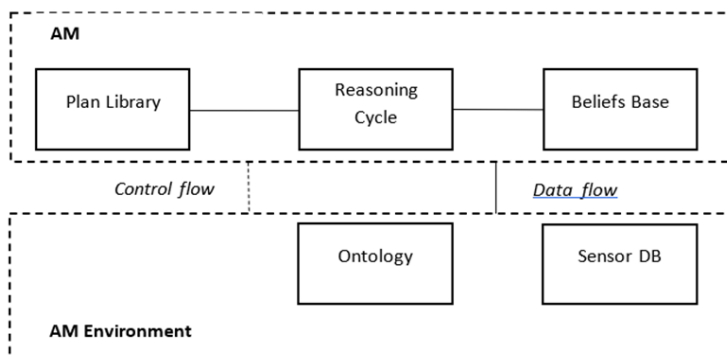
Платформата ACreM (Air Credible Monitoring) е агентоцентрична, хибридна и регионална [15]. Тя е агентоцентрична, защото двата основни компонента на платформата са имплементирани като агенти. Платформата е предназначена основно за използване в област Пловдив. За тази цел са подготвени специфични за региона параметри, които платформата използва в работата си. Архитектурата на платформата ACreM е дадена на фиг. 10.



Фигура 10. Архитектура на платформата ACreM

Air Monitor (AM) е личен асистент, предназначен да идентифицира и локализира различни аномалии, свързани със състоянието на въздуха, въз основа на нашите измервания. AM е разработен като BDI (Beliefs-Desires-Intention) агент в средата за разработка JaCaMo. Моделът BDI има своите корени във философската традиция за разбиране на практическото разсъждение у хората. Практическото разсъждение е ориентирано към действие – процес на търсене на какво да се прави. Практическото обсъждане е въпрос на претегляне на противоречиви съображения за и против конкуриращи се опции, зависещи от желанията, притесненията или ценностните преценки на агента. Като процес, практическото разсъждение включва две различни дейности. Първата, известна като обсъждане, е да се реши какво състояние на

нещата (цел) искаме да постигнем. Втората, наречена планиране, е да се реши как да се постигне това състояние на нещата (тази цел). Едно от предимствата на тази архитектура е способността за гъвкаво представяне и работа със средата на агента, което е от съществено значение за нашите цели. Друго предимство е, че АМ работи с проверен експертен опит за нашата област на интерес (в този случай замърсяване на въздуха). Това обстоятелство е едновременно и недостатък на агента ReAct, който работи със свободен текст, създаден и от непрофесионалисти като журналисти, политици и заинтересовани граждани. Поради това сме на мнение, че комбинирането на двата подхода е адекватен начин за постигане на целта на изследването, а именно търсенето на надеждна информация за състоянието на въздуха в град Пловдив. Фиг. 11 показва общата архитектура на АМ, разположена в работната му среда. Диаграмата показва основните компоненти на АМ и неговата среда.



Фигура 11. АМ архитектура

Базата от вярвания е дискретна структура, която моделира възприятието на агента за неговата среда. Индивидуалните вярвания са представени като предикати (в стила на езика за логическо програмиране Prolog). Структурата се състои от два базови компонента, статичен и динамичен. Всяка промяна в Базата от вярвания (изтриване или добавяне на вярване) задейства действително събитие. Този компонент се актуализира през зададен от нас интервал от време. Примерен сегмент от Базата от вярвания е даден на Фиг. 12.

```

/* Beliefs Base - represents the agent's initial knowledge: the sensor values from the Sensor DB and from Air Pollution
Ontology. CO2 level in parts per million (ppm), as well as PM2.5 level in micrograms per cubic meter (µg/m3) */
sensor_value(co2, current_value).
sensor_value(pm25, current_value).
co2_threshold(1000).
pm25_threshold(15).

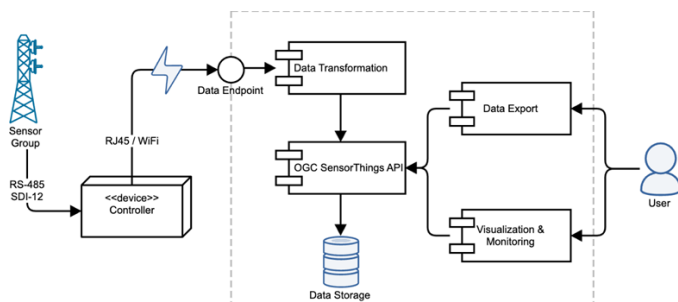
/* Initial goal causes the initial triggering_event to be generated, in this case to initialize and start the air quality
monitoring process */
!start_air_monitoring.

```

Фигура 12. Примерен сегмент от Базата от вярвания

В АМ средата са разположени две основни хранилища. Основните знания за разбиране и мониторинг на качеството на въздуха се съхраняват в онтология, наречена Онтология на замърсяването на въздуха, разработена в съответствие с глобалните насоки за качество на въздуха на Световната здравна организация, Изпълнителната агенция по околна среда на България и Системата за мониторинг на околната среда на община Пловдив предлагат изчерпателна информация за качеството на въздуха в България, включително измервания на ключови замърсители като PM10, PM2.5, SO₂, O₃, CO и NO₂, достъпни на официалния им уебсайт. Посочените данни служат като основа за изготвяне на заключения за качеството на въздуха чрез сравняването му с динамично въведени данни в базите данни. Второто хранилище е релационна база данни, която съхранява нашите измервания. Базите данни съдържат динамичните данни, получени от различни IoT възли или от данни в реално време, въведени от потребителите на платформата. Потокът от данни започва с неговото начало – измервания, извършени от устройства в групата сензори (фиг. 13). Записаните стойности впоследствие се четат и консолидират в един пакет данни от специално устройство за управление. Този контролер функционира като посредник, адаптирайки се между различните сензорни устройства, всяко със свой специфичен интерфейс, и по-широката система. На редовни интервали (обикновено на всеки 1 до 5 минути) контролерът предава пакета данни до крайната точка на данните на софтуерната система. При пристигането си пакетът данни се насочва към модул за трансформация на данни. Основната му функция е да раздели пакета данни на серия от отделни измервания, всяко от които е свързано със съответната му стойност и метрика. Имплементацията на модула за трансформация на данни използва потоците на Node Red. За всяко измерване, модулет за трансформация на данни инициира дискретна REST заявка към SensorThings API на OGC, който от своя страна съхранява измерването в хранилището за

данни. Имплементацията на Fraunhofer IOSB служи като гръбнак за SensorThings API. PostgreSQL, подобрен с разширението TimescaleDB, служи като определено хранилище за данни. След като данните са сигурно съхранени в хранилището за данни, извличането им става възможно чрез различни методи за четене на SensorThings API. Тези методи обхващат широк спектър от филтри, което позволява щателно персонализиране на извлечените данни. Приложения, като Grafana, могат да използват методите за четене на SensorThings API за наблюдение, визуализация или експортиране на данни. Grafana, например, разполага с удобен предварително конфигуриран източник на данни за безпроблемна интеграция с SensorThings API.



Фигура 13. Компоненти на архитектурата

Взаимодействието на AM със средата му се основава на концепцията за A&A (Артефакти и агенти). Air Monitor Copilot (AMCo) е агент-копилот с ReAct архитектура, който използва за изследване и оценка на данни за замърсяването на въздуха в района на град Бургас от външни източници. Рамката ReAct използва комбинация от декомпозиция на задачи, цикли на разсъждение и множество инструменти за разрешаване на проблеми. ReAct агентите от библиотеката LangChain могат да поддържат цялостен процес на обработка на заявки.

С въвеждането на платформата се цели да се допринесе за обективната оценка на качеството на въздуха и следователно за вземането на ефективни решения. В текущата версия, AMCo представя резултатите в неструктуриран формат (текст), който е лесен за четене от човешки потребител. При бъдещи взаимодействия с AM, обменът на данни също трябва да бъде в структуриран или полуструктуриран (напр. JSON) формат.

ЗАКЛЮЧЕНИЕ

Резултатите от проведеното изследване, представени в настоящата дисертация, могат да бъдат накратко обобщени както следва:

- Разработени са концепция, модел и архитектура на базов туристически екскурзовод.
- Създаден е работещ прототип на туристически екскурзовод.
- Демонстрирани са възможностите за адаптиране на прототипа за реални приложения в основната приложна област (туризъм) и в друга различна приложна област (опазване на околната среда).

Освен това могат да се направят следните изводи относно резултатите от проведеното изследване:

- Относно особености на виртуалния туристически гид – основният подход в създаването на виртуални туристически гидове е изготвянето на система, която да улесни механизма на традиционно търсене на информация и резервиране/закупуване на билети.
- Относно модел на персоналния екскурзовод и основни функционалности – виртуалният туристически гид е разработено мобилно приложение, което е налично за мобилни устройства с операционна система Android и уеб сайт (<https://exploring-gabrovo.com/>). Системата се състои от няколко части: мобилно приложение; публичен уеб сайт и панел за управление на съдържанието.
- Относно насоките за бъдещо развитие на виртуалния туристически гид – създаденият туристически гид ще спомогне работата на туристическия информационен център в Габрово, поради което е необходимо да бъде насочен към индивидуалния турист и да му предостави възможност за съставяне на индивидуален оптимален маршрут, според личните предпочитания. Виртуалният туристически гид трябва да се адаптира при промяна на околната среда, като смяна на метеорологичните условия, отклонение на туриста от зададения маршрут, задръстване по пътя и др.

СПИСЪК НА ЦИТИРАНИТЕ В ДИСЕРТАЦИЯТА

ПУБЛИКАЦИИ

**на Илия Илиев Неделчев,
докторант към катедра „Компютърни системи“**

1. Stoyanov, S.; Doychev, E.; Stoyanova-Doycheva, A.; Tabakova-Komsalova, V.; Stoyanov, I.; Nedelchev, I. A Regional Multi-Agent Air Monitoring Platform. Future Internet 2025, 17, 112. <https://doi.org/10.3390/fi17030112>
2. I. Nedelchev, V. Tabakova-Komsalova, I. Stoyanov, S. Stoyanov, V. Ivanova and T. Kazashka, "Supporting Digitization of a Cultural and Historical Heritage Platform," 2024 International Conference Automatics and Informatics (ICAI), Varna, Bulgaria, 2024, pp. 490-493, doi: 10.1109/ICAI63388.2024.10851643.

РЕФЕРИРАНА ЛИТЕРАТУРА

- [1] Божинова, М. Личев, Т. Тодорова, Л. Павлов, П. Георгиева, К. Съвременни измерения на управлението на туристическа дестинация България. Алманах научни изследвания, том 27, 2019, с. 7-22
- [2] Kanwel, S. Lingqiang, Z. Asif, M. Hwang, J. Hussain, A. Jameel, A. The Influence of Destination Image on Tourist Loyalty and Intention to Visit: Testing a Multiple Mediation Approach. Sustainability 2019, 11, 6401. <https://doi.org/10.3390/su11226401>
- [3] Szostak, D. Spatial and Infrastructural Conditions of Functioning of the Contemporary Tourism Market – Analysis of Selected Issues. European Research Studies Journal. Volume XXVI, Issue 2, 2023, pp. 337-347
- [4] Pai, C. Y. Liu, S. Kang, A. Dai. The role of perceived smart tourism technology experience for tourist satisfaction, happiness and revisit intention. Sustainability, 12 (16) (2020), p. 6592, 10.3390/su12166592
- [5] Радулов, Н. Сигурност 4.0. София, Научнотехнически съюз по машиностроене, 2019, с. 72
- [6] Антонова, К. Иванова, П. Роботизацията и изкуственият интелект – нов модел на сътрудничество и/или взаимодействие между машините и хората на работното място. Човешки ресурси & Технологии. Бр. 1, 2021, с. 65
- [7] Alberto M. Flutter Complete Reference: Create beautiful, fast and native apps for any device. Independently published, 2020. p. 5-10
- [8] Dino E. Programming ASP.NET Core, First edition, fast and native apps for any device. Microsoft Press, 2018. p. 15-16
- [9] Matthias B. RESTful API Design: 3 (API-University Series). CreateSpace Independent Publishing Platform, 2016. p. 50-53
- [10] Nigel P. Docker Deep Dive: Zero to Docker in a single book. Independently published, 2020. p. 20-21
- [11] Hans-Jurgen S. Mastering PostgreSQL 13: Build, administer, and maintain database applications efficiently with PostgreSQL 13, 4th Edition. Packt Publishing, 4th ed. Edition, 2020. p. 10-13
- [12] I. Nedelchev, V. Tabakova-Komsalova, I. Stoyanov, S. Stoyanov, V. Ivanova and T. Kazashka, "Supporting Digitization of a Cultural and Historical Heritage Platform," 2024 International Conference Automatics and Informatics (ICAI), Varna, Bulgaria, 2024, pp. 490-493, doi: 10.1109/ICAI63388.2024.10851643.
- [13] Rafalimanana, H. J. L. Razafindraminta1, S. Cherrier, T. Mahatody, L. Geogrg, V. Manantsoa, Jason-RS, a Collaboration between Agents and an IoT Platform, International Workshop

- [14] Soliman, M. Guetl, V. Evaluation of intelligent agent frameworks for human learning. 14th International Conference on Interactive Collaborative Learning, Piestany, Slovakia, 2011, pp. 191-194, doi: 10.1109/ICL.2011.6059574.
- [15] Stoyanov, S.; Doychev, E.; Stoyanova-Doycheva, A.; Tabakova-Komsalova, V.; Stoyanov, I.; Nedelchev, I. A Regional Multi-Agent Air Monitoring Platform. Future Internet 2025, 17, 112. <https://doi.org/10.3390/fi17030112>.