**ТЕХНОЛОГИЧНО УЧИЛИЩЕ ЕЛЕКТРОННИ СИСТЕМИ**

**към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ**

**ДИПЛОМНА РАБОТА**

Тема: Сердика – Приложение с добавена реалност

Дипломант: Научен ръководител:

*Димитър Кацаров Росен Петков*

СОФИЯ

2016

**TЕХНОЛОГИЧНО УЧИЛИЩЕ**



**ЕЛЕКТРОННИ СИСТЕМИ**

**към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ**

Дата на заданието: 10.11.2015 г. Утвърждавам:..............................

Дата на предаване: 10.02.2016 г. /проф. д-р инж. Т. Василева/

**ЗАДАНИЕ**

**за дипломна работа**

на ученика Димитър Петров Кацаров 12А клас

1.Тема: Сердика – Приложение с добавена реалност

2.Изисквания:

2.1. Разпознаване на обект от камера на телефон

2.2. Добавяне на информация, медии към обекта.

3.Съдържание

3.1. Обзор

3.2. Същинска част

3.3 Приложение

Дипломант :...........................................

Ръководител:..........................................

/гл.ас. инж. ххххххххх/

Директор:................................................

/ доц. д-р инж. Ст. Стефанова /

**УВОД**

В настоящата дипломна работа се разработва приложение с добавена реалност за интересни забележителности в София на име SerdikaAR. Проектът е базиран на съществуващото приложение “Invisible Serdika”.

Обогатена реалност (Augmented reality) е изглед към нашия физически свят, подобрен или изменен със звук, видео, графика или данни за позиция. Изменяването се случва на момента. С помощта на високо развитата технология за разпознване на обекти и добавянето на компютърно генерирана медия, света около потребителя става интерактивен и дигитално манипулиран.

Определението на Роналд Азума е едно от по-насочените описания. То покрива подмножество от оригиналните идеи на добавената реалност, но достига до представящо цялата област на добавената реалност. Според Азума тя представлява среда, която включва виртуална реалност и елементи от реалния свят. Той дефинира системата за добавената реалност като система, която комбинира реално и виртуално, интерактивна в реално време и работи в три измерения. Това определение често се използва в научната литература.[1]

Целта на SerdikaAR е да улесни и да направи по интересен и интерактивен начин за разглеждане на забележителностите в столицата. В него се включват три главни опции:

1. Визуализиране на маркери в реално време свързано с географските местоположения на обекти на интерес.
2. Визуализация на триизмерни модели върху физическа карта на София.
3. Разпознаване на маркер с цел показване на информация свързана с обекта.

Към приложението се разработва сайт, предназначен да улесни въвеждането на информация към базата. Неговите функционалности са :

1. Показване на информация за приложението с цел да научи

потребителите как се използва и да рекламира приложението.

2. Въвеждане на още географски кординати чрез Google Maps Api

**ПЪРВА ГЛАВА**

**ПРЕГЛЕД НА ПОДОБНИТЕ ПРОДУКТИ НА „SERDIKA-AR“ И ТЕХНОЛОГИИ ЗА РАЗРАБОТКАТА МУ**

**1.1. Преглед на подобни продукти**

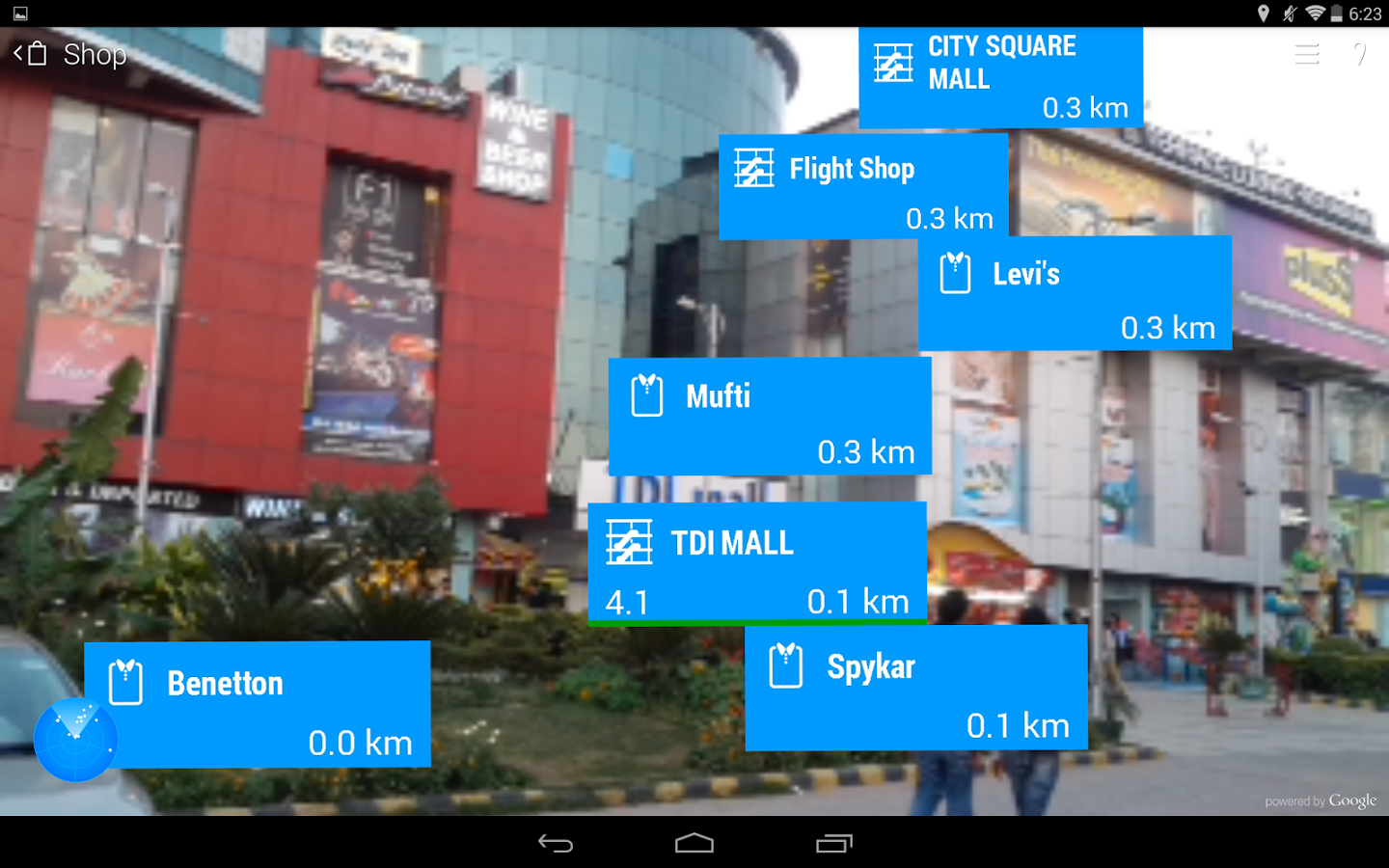
**1.1.1. Here City Lens**

****

Here city lens е софтуер за добавена реалност, който дава динамична информация, чрез дисплей на камерата на телефона, за околността на потребителите като магазини, ресторанти и точки на интерес, показани като виртуални знаци, насложени върху или над сгради. Една търговска Бета версия е издаден през 2013 г.

Има безплатена гласова навигация, която може да се използва без интернет връзка, използвайки предварително инсталирани карти.[2]

**1.1.2 World around me**



Светът около мен е идеалният инструмент за пътници, туристи или местни хора, които искат да знаят всичко за това, което е около тях. Открийте места като насочите телефона си около теб, и да се чувстват като у дома си на всяко място в света! Нека WAM ви води на вълнуващо пътешествие към местата около вас.

Тук са само някои от местата, които WAM ще идентифицират, локализират и показват мнения за:

- Ресторанти | Барове | кафенета

- Банките | банкомати

- Киносалони

- бензиностанциите

- Метростанции | Гари | Автобусни спирки | Такси Щандове | Летища

- Болниците | Стоматологични клиники

- Аптеки

- Хотели

- Търговски центрове | Grocers |

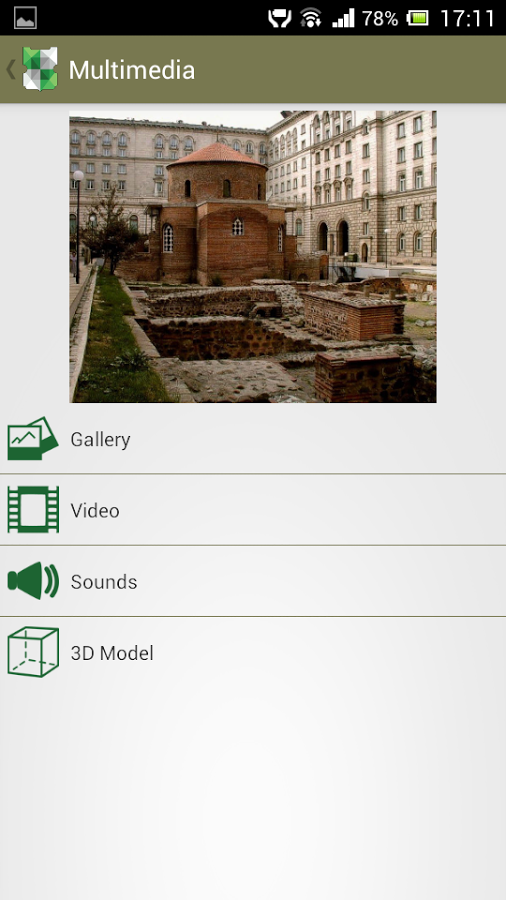
магазини за дрехи

- Книжарници | Магазини за обувки

- Църкви | Джамиите | синагоги | Храмовете

WAM съчетава най-модерните технологии за добавена ралност с изчистен красив интерфейс, който добавя полезна и вълнуваща информация за всички места около вас. WAM показва имена, оценки и разстояние в една безпроблемна и интуитивна гледна обогатена реалност. Още един бутон и може да получите достъп до подробни прегледи, работното време, посоки и данни за контакт.[3]

**1.1.3 Invisible Serdica**

****

„Невидимата Сердика“ е безплатно приложение за мобилни телефони, което дава възможност на потребителите да научат повече за скритите и най-впечатляващи исторически и архитектурни обекти в София, столицата на България, посредством изображения, видео материали, 3D модели, интересни истории и легенди за подбраните обекти, различни от стандартната информация в пътеводителите. Като начало бяха подбрани 5 ключови обекта:

1/ Църква „Св. София“

2/ Ротонда „Св. Георги“

3/ Античен амфитеатър

4/ Ларгото (включ. обектите около и под него)

5/ Гробниците под жълтите павета

Като източници за по-голямата част от съдържанието в приложението са използвани стари книги и поместените в тях уникална информация, гравюри, изображения, чертежи и др. Идеята на приложението е да разкрие и направи достъпна за по-широк кръг потребители тази интересна, но малко известна информация и по този начин да направи най-вдъхновяващите обекти в София (чието римско име е било „Сердика“) още по-привлекателни.

Друга, също така важна, цел на приложението е да популяризира НАБИС каталога като включва линкове към книги и различни публикации, поместени в него и свързани с избраните пет обекта.

Приложението „Невидимата Сердика“ може да е полезно не само за ползвателите на НАБИС каталога и библиофилите, но и за туристи и посетители на София, както и за ученици, студенти и преподаватели.[4]

**1.2 Преглед на подобни технологии за разработка**

**1.2.1 Unity -** Юнити е междуплатформен игрови двигател, разработен от Юнити Технолоджис и използван за разработка на видеоигри за компютри, конзоли, мобилни устройства и сайтове. В началото е обявен само за OS X, на световната конференция за програмисти на Епъл, през 2005 г. От тогава е разширен за повече от петнадесет платформи. В момента е основният инструмент за разработка на софтуер (SDK) за Wii U.

Професионалната версия на Юнити е достъпна срещу определена такса, а личната няма такса – свободно е за ползване от всеки човек или компания с по-малко от 100 000 долара годишен приход. С пускането на Юнити 5.0 на 3ти Март 2015 г., Юнити Технолоджис предостави целия двигател за свободно ползване, с всички подробности, изходен код и премийна поддръжка. Юнити е признат заради възможността си да предоставя игри за множество платформи.

Пет версии на Юнити са били пуснати досега. През 2006 г. на WWDC Trade show, Епъл обявява Юнити за първи подгласник в категорията за най-добро използване на Mac OS X графика.[5]

**1.2.2 Unreal Engine** - Unreal Engine e игрови двигател, разработен от Epic Games. За пръв път е демонстриран през 1998 година в играта Unreal. Въпреки че е разработен предимно за шутъри от първо лице, той е използван успешно в редица игри от различни жанрове като стелт, MMORPG и RPG. Софтуерният код на Unreal Engine е написан на езика C++, което му осигурява висока степен на портативност. Днес се използва от много разработчици на игри.

Текущата версия е Unreal Engine 4, предназначена за технологията DirectX 11 и 12 на Майкрософт(за Microsoft Windows, Xbox One и Windows RT); OpenGL (за OS X, Linux, PlayStation 4, iOS, Android, Ouya и Windows XP); и JavaScript/WebGL (за уеб браузъри, поддържащи HTML5).[6]

**1.2.3. Java (Android Studio)** - Java е обектно-ориентиран език за програмиране. Кодът, написан на Java, не се компилира до машинен код за определен процесор, а до специфичен за езика код, наречен байт код. Поради това за изпълнението на програма, написана на Java, е необходима т. нар. виртуална машина

Подобна реализация има своите предимства и недостатъци. Сред главните предимства са:

лесната преносимост между различните платформи (софтуерни или хардуерни) – веднъж написана и компилирана, една Java-програма може да бъде стартирана на компютри независимо от архитектурата или от операционната им система. За целта е необходимо само да се инсталира виртуална машина за съответната платформа;

допълнителните действия, извършвани от виртуалната машина като освобождаване на паметта от обекти (класове) които не се използват (Garbage collector), проверка за размерността на масивите;

възможността за контрол на правата на потребителя на ниво виртуална машина

първоначално заделяна на heap, част от паметта резервирана за джава

висока степен на сигурност поради факта, че програмистите не работят директно с паметта и др.

Главен недостатък е необходимостта от допълнителни ресурси (под формата на процесорно време и памет) за изпълнението на самата виртуална машина.

Съществуват и компилатори, които превеждат байт кода до машинен код. Недостатъка при използването им е, че тогава програмата не може да се стартира на различни платформи.

Ранните версии на виртуалните машини страдат от слаба производителност поради необходимостта от компилиране на байт кода до машинен код. Този проблем е решен в значителна степен след въвеждането на JIT компилатора във виртуалната машина.[7]

**1.2.4 PHP -** PHP е скриптов език върху сървърната (обслужваща) страна език с отворен код, който е проектиран за уеб програмиране и е широко използван за създаване на сървърни приложения и динамично уеб-съдържание. Автор на езика е канадецът от датски произход Размус Лердорф. PHP е рекурсивен акроним от PHP: Hypertext Preprocessor (като в самото начало има значение, дадено от създателите му, на Personal Home Page). Пример за PHP приложение е МедияУики – софтуерът, използван от Уикипедия.[8]

**1.2.5. Vuforia – „**Augmented Reality Software Kit“ за развитие (SDK) за мобилни устройства, която позволява създаването на приложения за обогатена реалност. Той използва технологията „Computer Vision“ да разпознае и да следи равнинни изображения и прости 3D обекти, като например кутии, в реално време. Тази възможност за регистрация на изображението дава възможност на разработчиците да се позиционират и ориентиране на виртуални предмети, като 3D модели и други медии, във връзка с реалния свят изображения, когато те се гледат през камерата на мобилно устройство. Виртуалният обект след това проследява позицията и ориентацията на изображението в реално време, така че перспективата на зрителя върху обекта кореспондира с тяхната гледна точка на целта за снимката, така че да изглежда, че виртуалния обект е част от реалния свят сцена.

**ВТОРА ГЛАВА**

**ИЗИСКВАНИЯ КЪМ ПРОГРАМНИЯ ПРОДУКТ. СРЕДСТВА ЗА РАЗРАБОТКА. ОПИСАНИЕ НА АЛГОРИТЪМА**

***2.1. Изисквания към продукта***

**2.1.1. Функционални изисквания към приложението за обогатена реалност**

**-** Използване на камерата на телефона

**-** Откриване на GPS координати

- Разпознаване на ориенатация на телефона

- Свързване с база данни

- Разпознаване на изображение (шаблон)

- Добавяне на обекти

***2.2. Избор на езика за програмиране и програмните средства***

За разработка на мобилното приложение избрах “Unity”. С него лесно се ориентира в 3D сцени и се добавят модели. Поддържа множество платформи и работи добре с друг от инструментите, които избрах (Vuforia). Unity поддържа Microsoft Visual Studio и C#. Това прави разработването на него по лесно и комфортно за мен.

Първоначалната разработка беше на Android Studio със OpenCV. Към Android Studio също лесно може да се свърже “Vuforia”, но визуализирането на 3D модели не беше толкова лесно. Има предимства на този метод, но в този случай използването на енджин ми спести много време и трудности.

За свързване на Unity със сървара избрах PHP, с което да взимат информация от базата. Методите, които Unity поддържа направиха използването на тези скриптове лесно. Избрах този начин, защото и преди съм правил мобилно приложение, което се свърза с базата чрез такива скриптове.

Използвах SQL база качена на сървар създаден чрез xampp, където е и сайта. Свързвам се с нея с IP и порт отворен на рутера.

За сайта е използван готов шаблон[9] пренаписан и обработен да изпълнява нуждите на приложението. Използвани са HTML, CSS и Javascript и използва Google Maps API са избиране на кординатите.

***2.3. Структура на приложението***

Приложението има 5 сцени:

* Menu
* Explore
* Map
* LearnMore
* Information

Сайта има 6 секции:

* Home
* Blog
* About
* Locations
* Contact

**2.3.1.1. Menu**

Това е главното меню на приложението. В него се съдържа герб, поле за информация и бутони, които променят полето за информация да съответства на това, което е избрано.



Бутоните отварят съответните платна:

Explore MapView LearnMore



**2.3.1.2. Explore**

Тази сцена е главната функционалност на приложението. При стартиране се зареждат от базата всички шаблони, които ще се визуализират. Инстанцира се предварително направен “prefab” с инфорамцията и се зареждат на съотвентния ъгъл спрямо географските кординати.

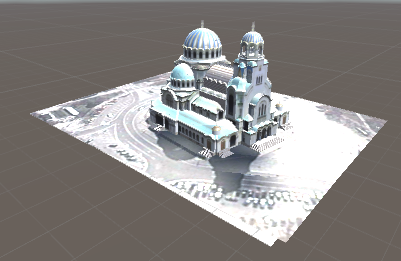
Включва се камерата на телефона, gyroscope и GPS. Всички шаблони се завъртат да гледат към камерата за да се виждат изцяло от потребителя. Камерата в сцената се върти спрямо ориентацията на телефона и така визуализира, когато телефона е обърнат в посока на шаблон. При натискане на този шаблон се отваря сцената за информация.



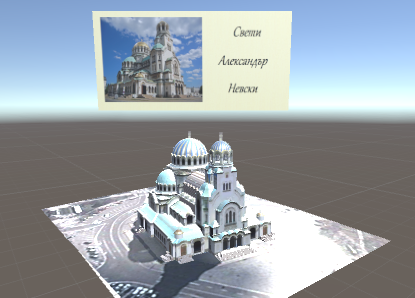
Под шаблона се показва разстоянието то обекта в метри.

**2.3.1.3. Map**

Тази сцена използва Vuforia, за да разпознава част от картата и да визуализира 3D обект на сградата, която съответсва на това място.



При натискане на този обект се появява шаблон върху него, който следи камерата. При натискането му се отваря сцената за информация.



**2.3.1.4. LearnMore**

Тази сцена е за бързо достъпване на информацията предоставена от приложението, чрез разпознаване на изображение или QR код свързан със съответния обект.



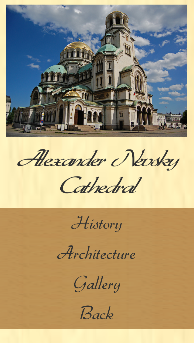
**2.3.1.5. Information**

Тази сцена е да показва на потребителя информация за обект взета от базата. Тя е обща за всички обекти. Данните се зареждат от базата за съотвентния обект, за който се е отворила сцената.

Съдържа секция с името. снимка на обекта и бутони за съответните секции:

History(История), Architecture(Архитектура) и Galley(Галерия).

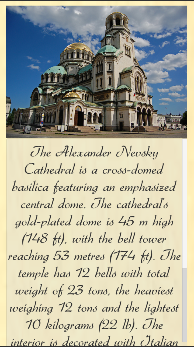
Главната секция



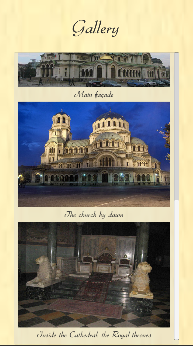
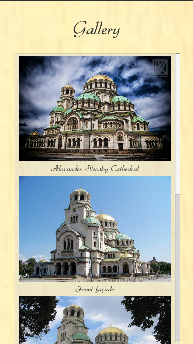
History, което съдържа историята на обекта със “scroll”



Architecture, което съдържа ахритектурата на обекта със “scroll”

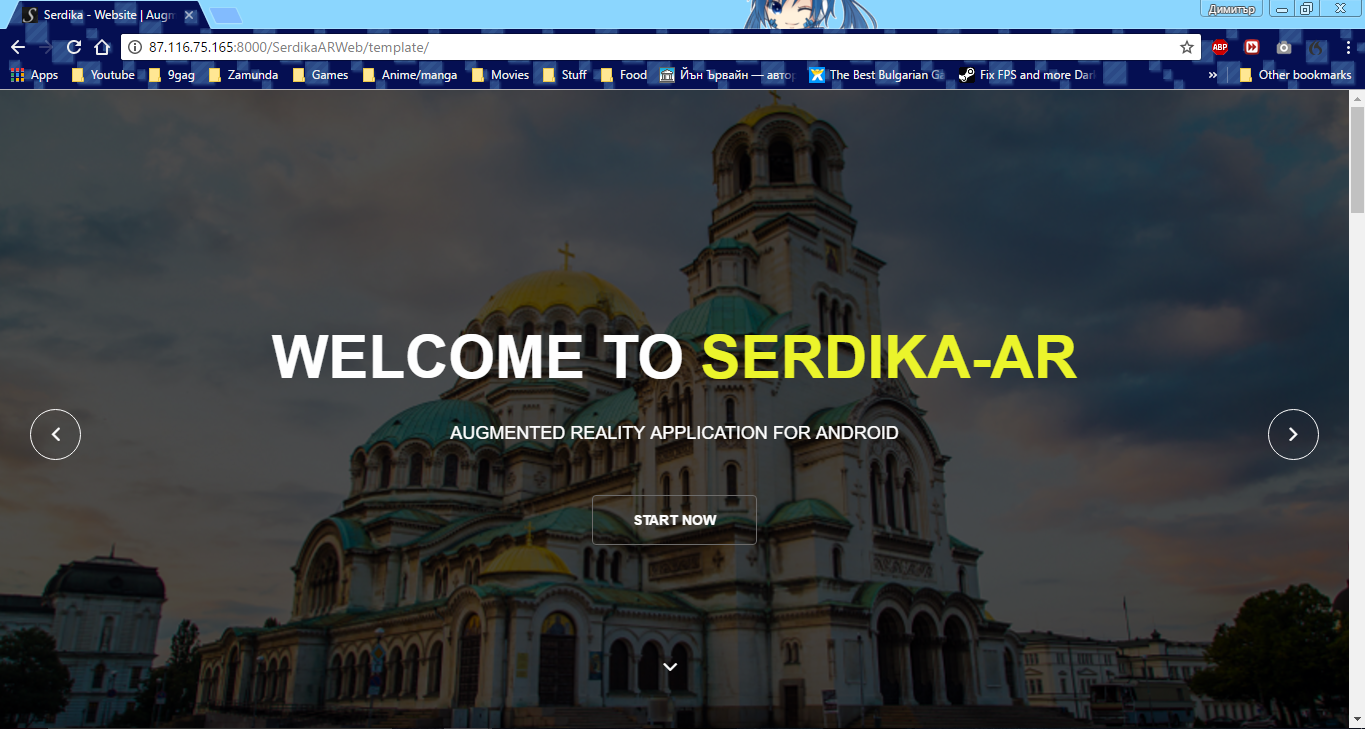


Gallery, което съдържа различни снимки на обекта с кратко описание към тях със “scroll”.

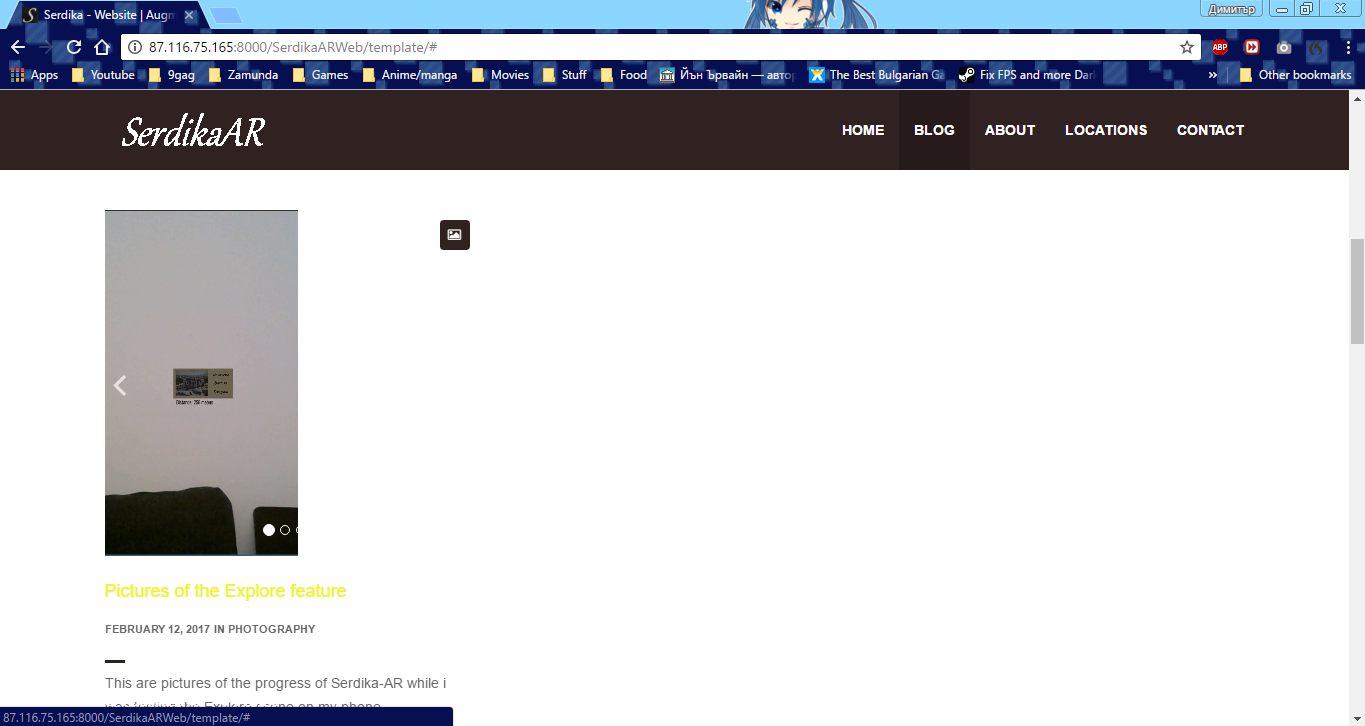


**2.3.2.1. Home**

Това е главната секция на сайта. Съдържа описание, няколко снимки и навигация.

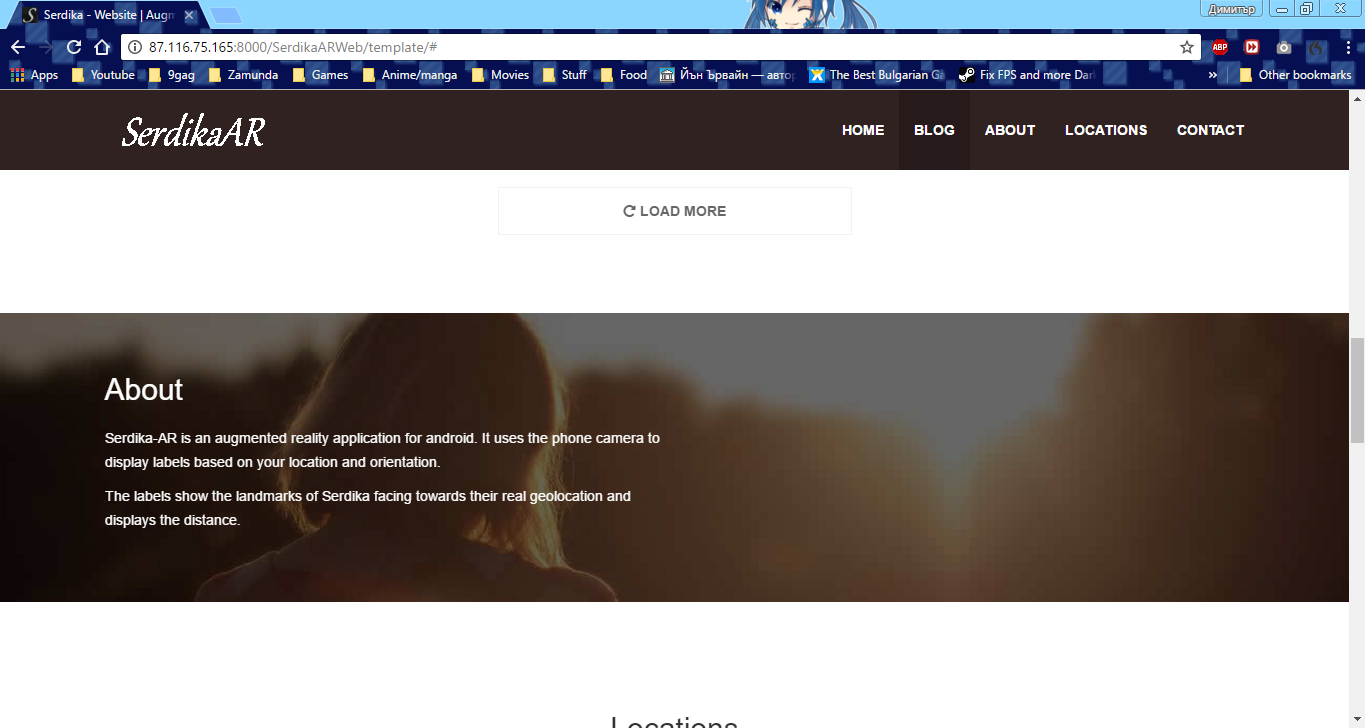


**2.3.2.2. Blog**

Тази секция съдържа обикновен блог за добавяне на снимки и информация за приложението.

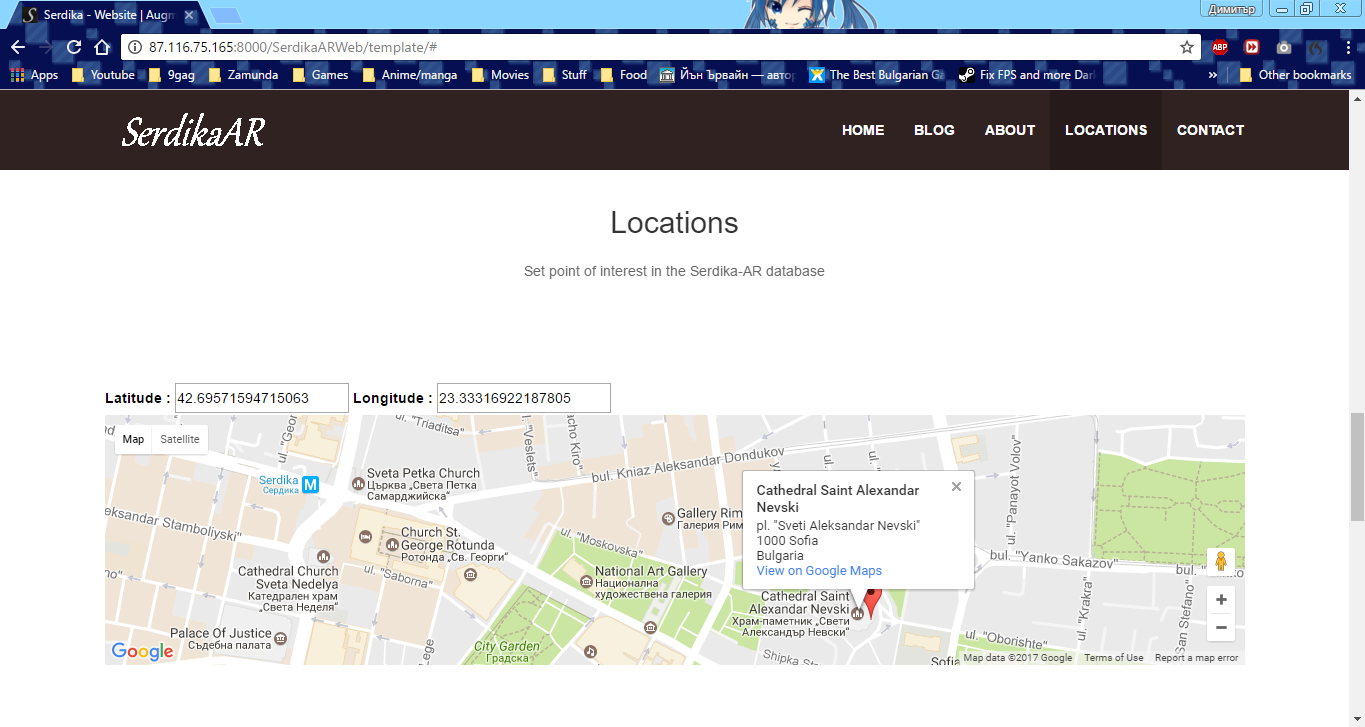
**2.3.2.3. About**

Тази секция разказва за проекта.



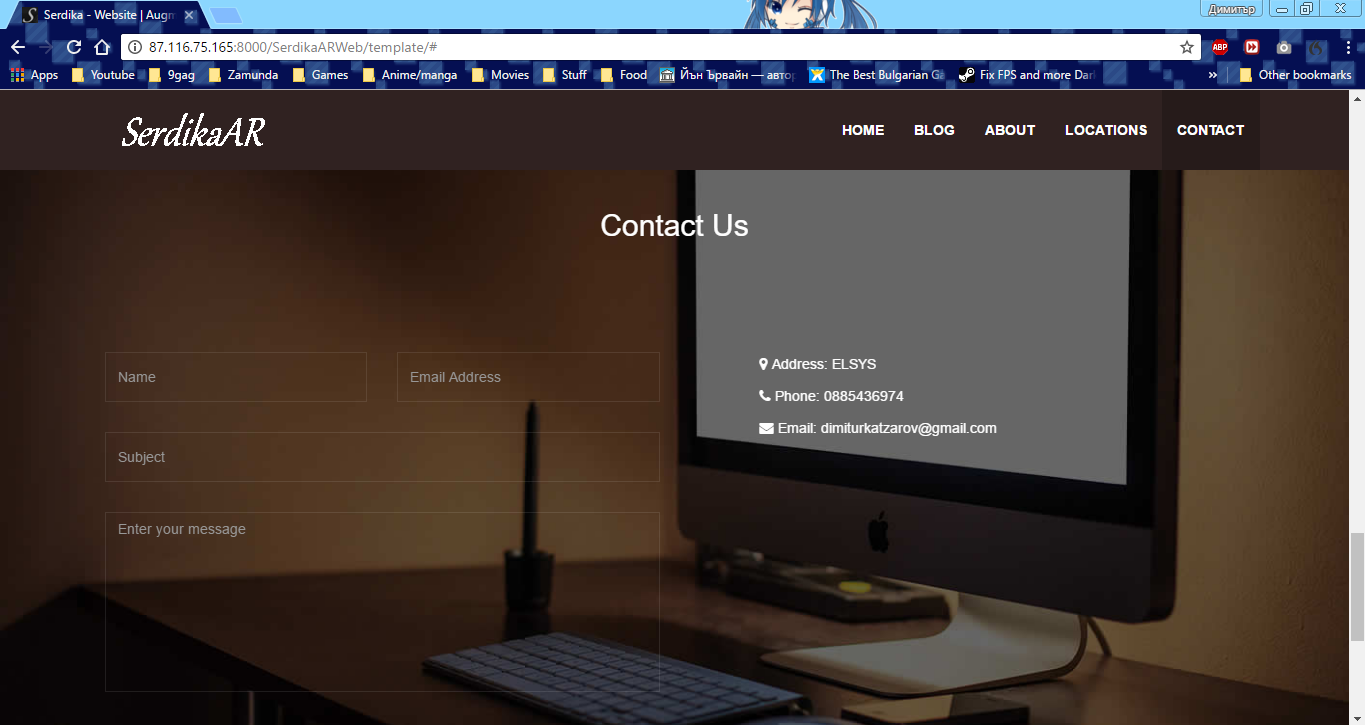
**2.3.2.4. Locations**

Тук се създават нови шаблони като се избира позицията им на картата и се записват в базата.

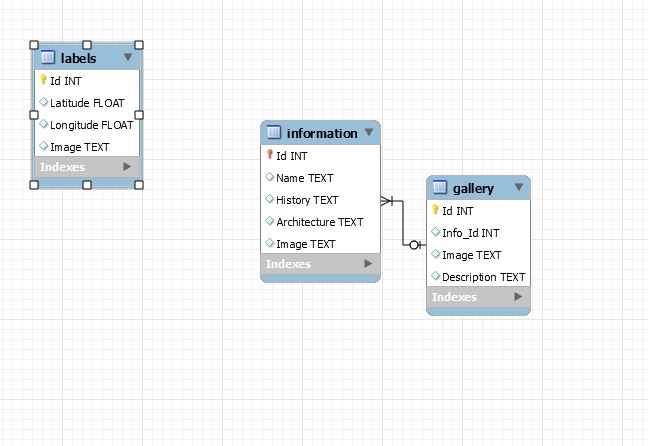


**2.3.2.5. Contact**

Тук е показана информация за връзки.



**2.4. Структура на базата данни**



Разделена е на 3 таблици:

* “Labels” – Съдържа данни за шаблоните, които се виждат в Explore сцената:
  + Latitude : географски кординати
  + Longitude : географски кординати
  + Image : url към изображението на шаблона намиращо се на сървара
* “Information” – Съдържа информация за обектите, която се визуализира в сцената Information :
  + Name : името на обекта
  + History : текст, който разказва за историята на обекта.
  + Architecture : текст, който разквазва за ахритектурата на обекта.
  + Image : url към изображението на обекта намиращо се на сървара.
* “Gallery”- Съдържа снимки на обектите, които се визуализират в секция Gallery на сцената Information.
  + Info\_Id : число което идентифицира към кой обект е съответната снимка.
  + Image : url към изображението намиращо се на сървара
  + Description : кратко описание на изображението

**ТРЕТА ГЛАВА**

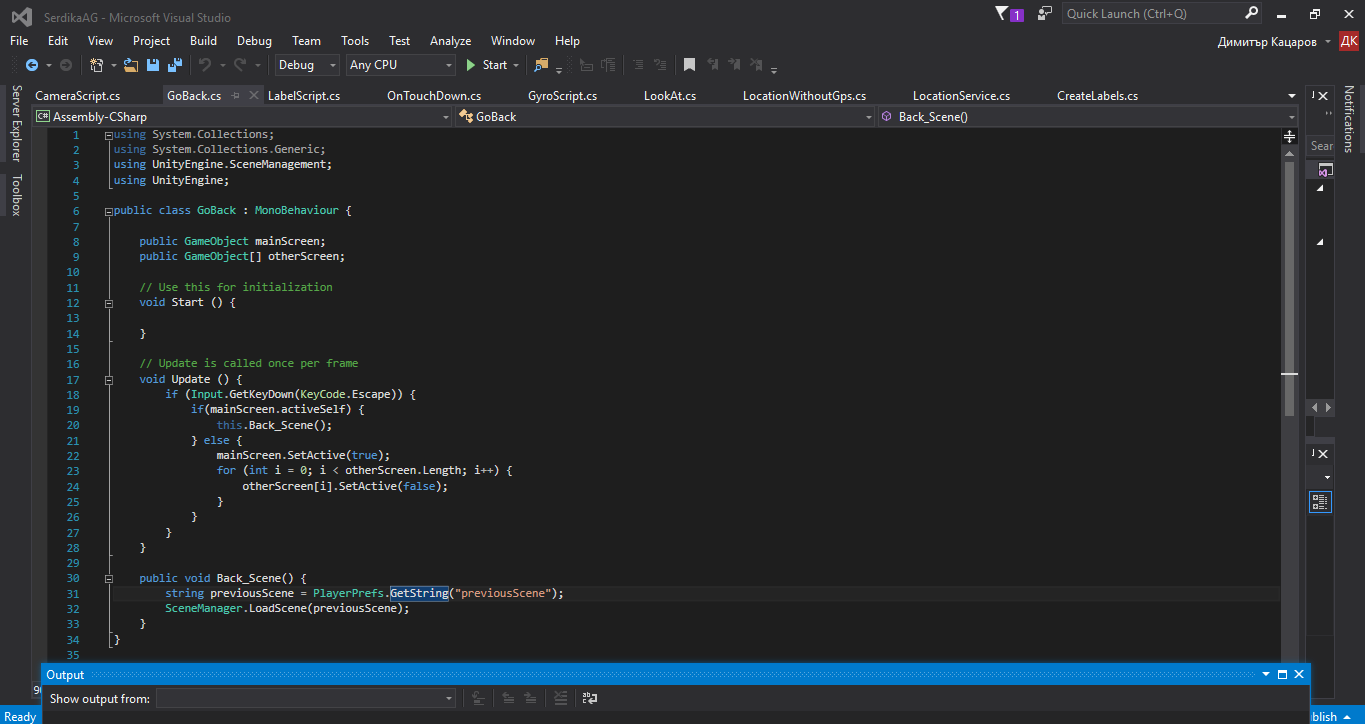
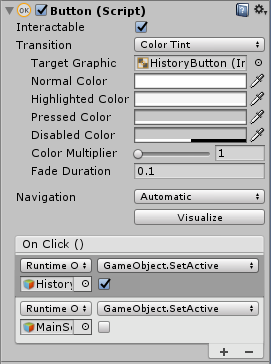
**ОПИСАНИЕ НА НАЧИНА НА РЕАЛИЗАЦИЯ НА ПРОЕКТА**

**3.1. Сцени**

**3.1.1. Menu**

Тази сцена е първата, която вижда потребителя. Тя е направена да изглежда добре и лесна за разбиране. С нея се достъпват другите сцени и части на приложението с бутони. Тук се използва енджина, за да се оформи дизайна на менюто, така че да изглежда добре на всякакви екрани. Има лек фон, няколко текстури с различни цветове и няколко изображения.

Контролирането на сцените и секциите се получава с евенти на Unity и един скрипт:



"Фигура 1. GoBack.cs"

При натискане на някой от бутоните този скрипт се активира и ако съдържанието на съответния бутон не е видимо, то се показва на място на другата секция. Ако съдържанието му е видимо, при натискане то се скрива за да се покаже главното съдържание.

При натискане на back бутона, ако е отворена главната секция, приложението се изключва, в противен случай се скриват всички секции и се отваря главната.

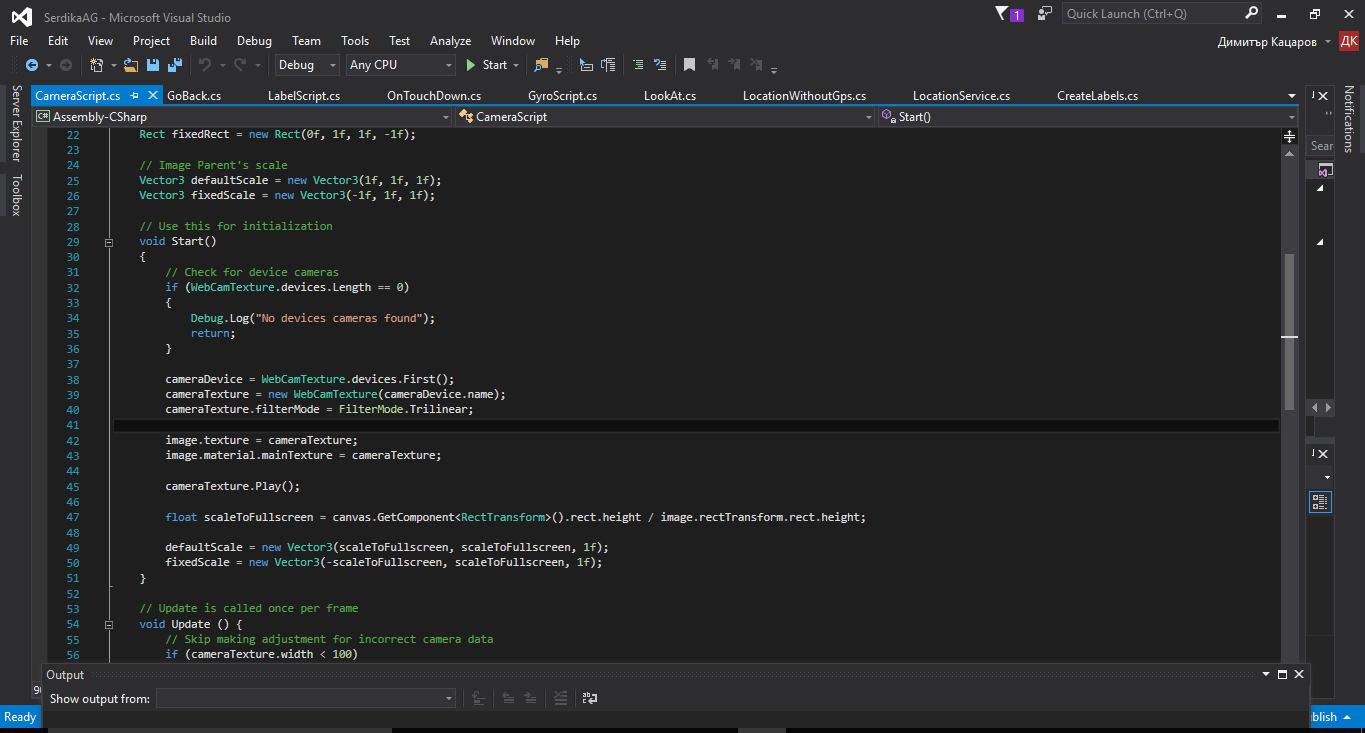
При натискане на Exit бутона, приложението се изключва.

В този скрипт се намира и метода “Back\_Scene()”, който зарежда от “PlayerPrefs” стринг с името на предишно отворената сцена и я отваря.

**3.1.2. Explore**

CameraScript: Тук се зарежда камерата на устройството и казва на картината да се обновява.

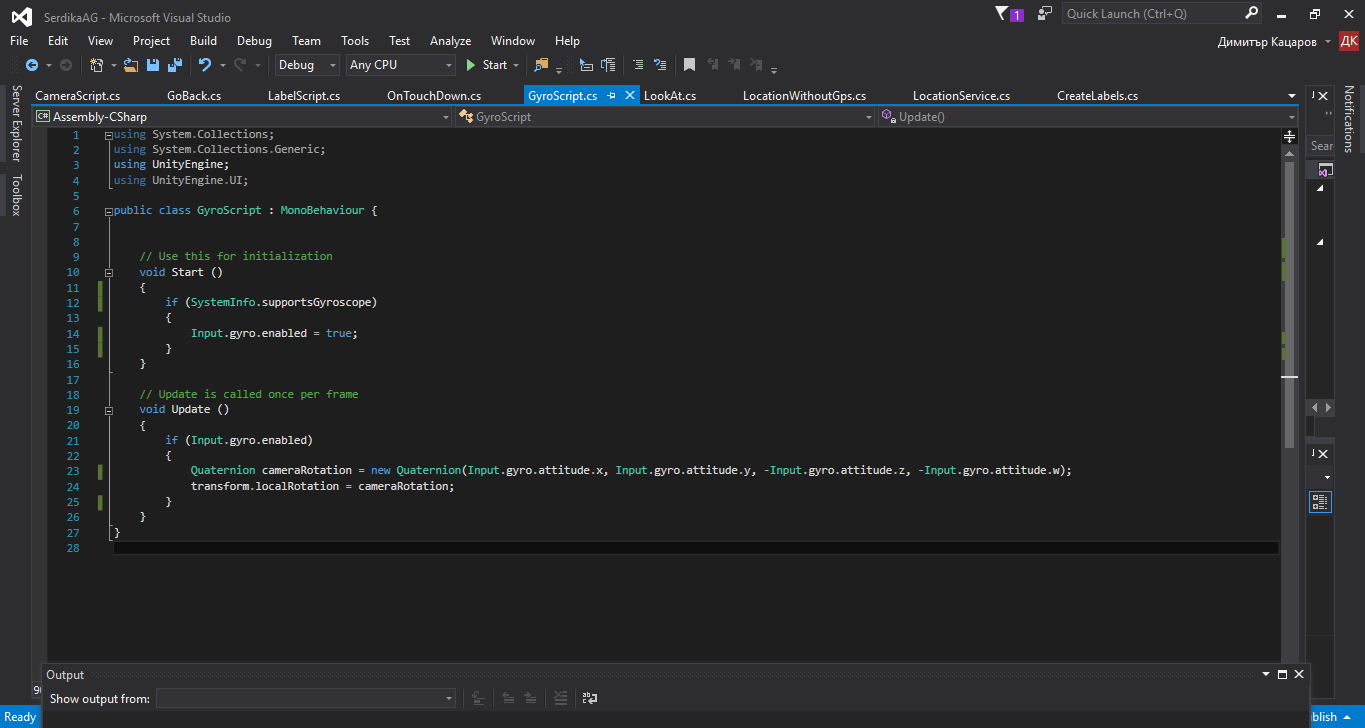
После се смята коефициента нужен наизображението да се разтегне за да запълва целия екран.



"Фигура 2. CameraScript.cs"

GyroScript: Проверява дали устройството има физически gyroscope и ако има го активира.

Всеки фрейм засича ориентацията на телефона и наглася ротацията на камерата в Unity да отговаря.



"Фигура 3. GyroScript.cs"

CreateLabels: Този скрипт изпълнява PHP скрипт, който прави заявка към сървара. Със стойностите, които получава инстанцира нов обект от “prefab” на име label за всеки ред от базата.

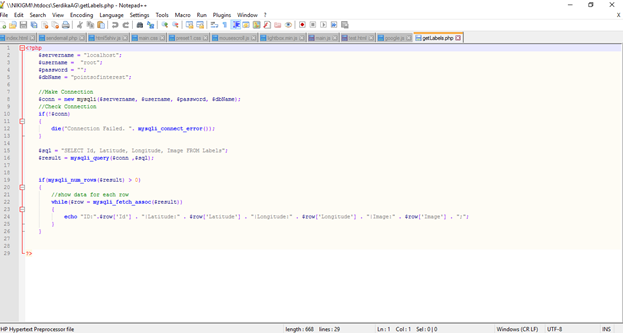


"Фигура 4. CreateLabels.cs"

От базата се взимат географски кординати, с които по късно се смята в каква посока се намират от теб и url към изображението което е качено на сървара. След това се изчаква изтеглянето на изображението и се слага като texture на новия инстанциран обект.

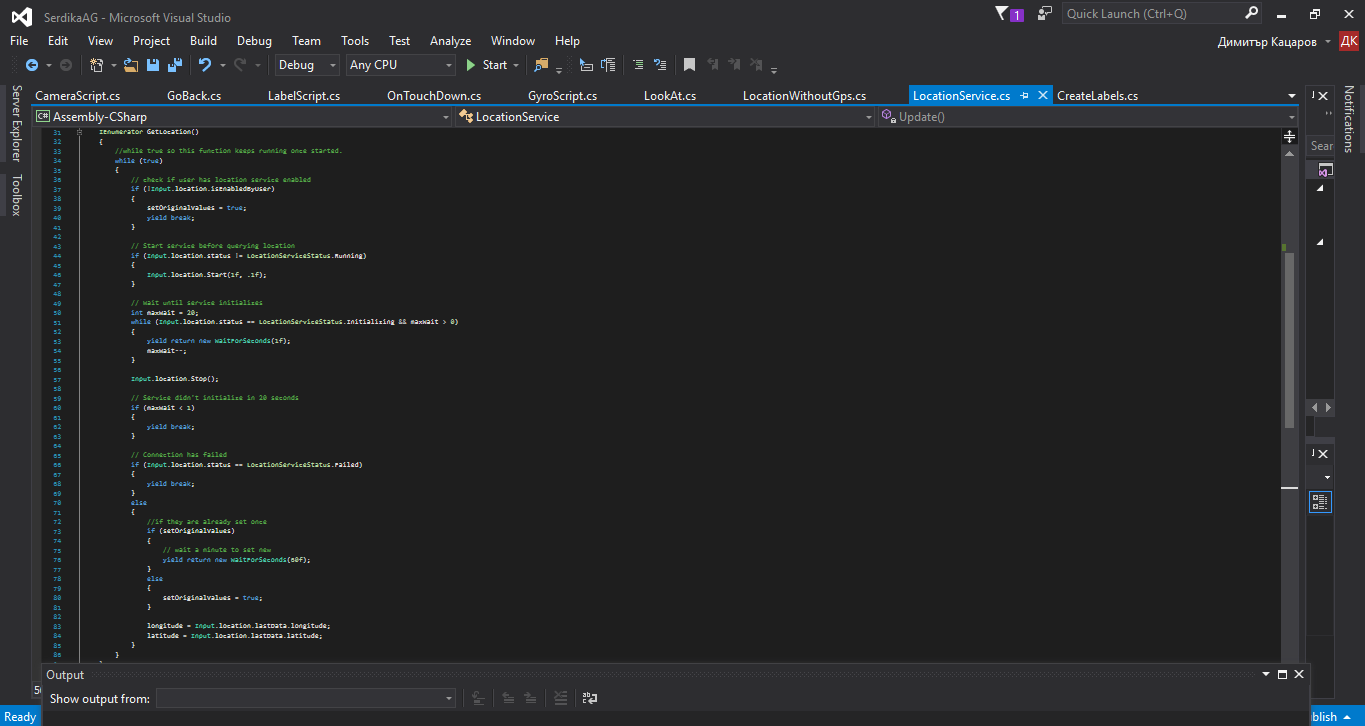
GetLabels.php : изпълнява заявка към базата, която взима всички въведения в таблицата labels с параметрите: Id, Latitude, Longitude, Image;

Изкрава всички върнати редове с echo, което позволява на Unity да ги получи.

****

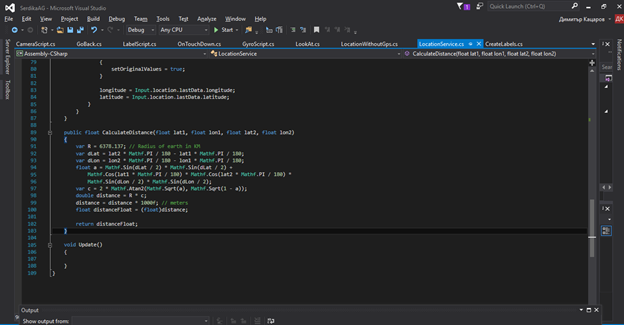
"Фигура 5. GetLabels.php"

LocationService: Стартира се Coroutine, който взима географското положение на потребителя всяка секунда. В него се проверява дали потребителя е позволил на приложението да му използва кординатите. Ако е позволил се опитва да вземе кординатите от GPS-а на телефона и чака до 20 секунди за отговор.



"Фигура 6. LocationService.cs"

В този скрипт се намира и метод който смята дистанцията в метри между две

географски точки, за да показва разстоянието на потребителя до обекта. 

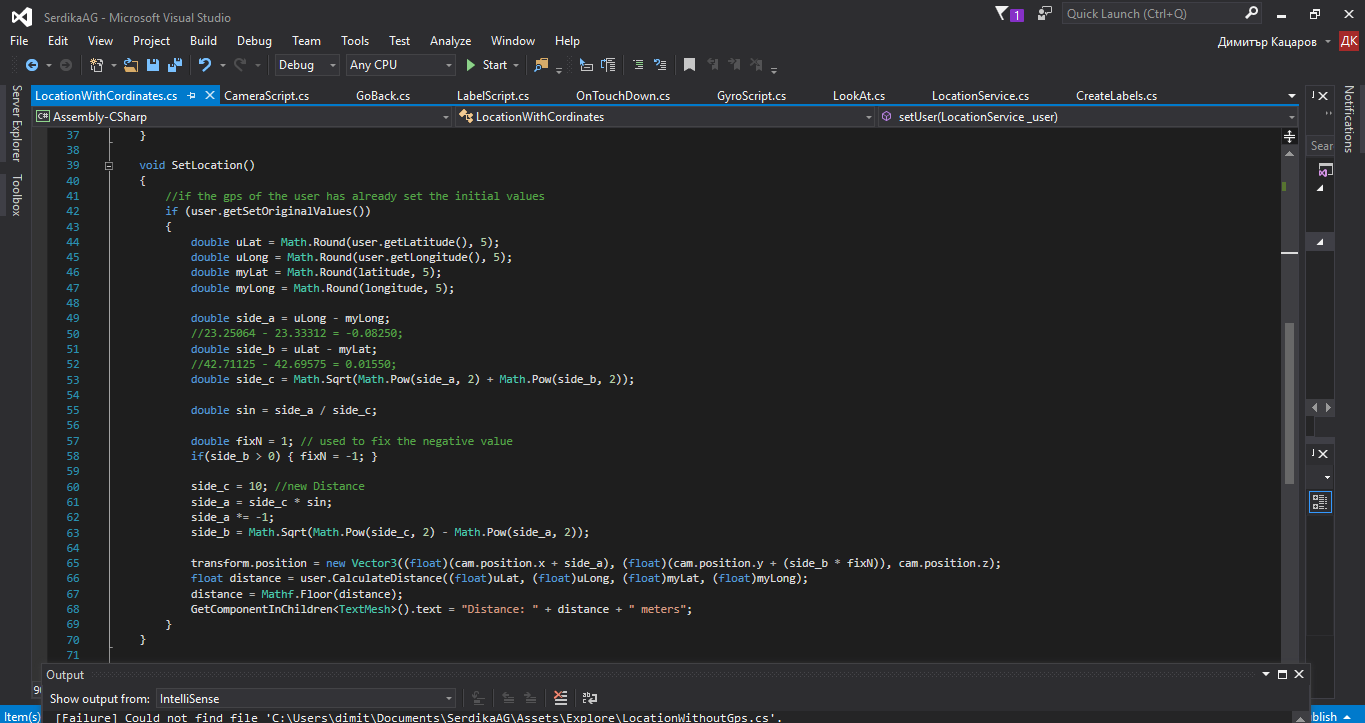
"Фигура 7.LocationService.cs : CalculateDistance()"

LocationWithCordinates : Този скрипт изчаква да бъдат взети кординатите на потребителя и след това всеки фрейм смята в каква посока да сложи обекта.

С двете географски точки се смятат двете страни на триъгълник от кординатната система. С тях се намира хипотенузата. Намираме синуса.

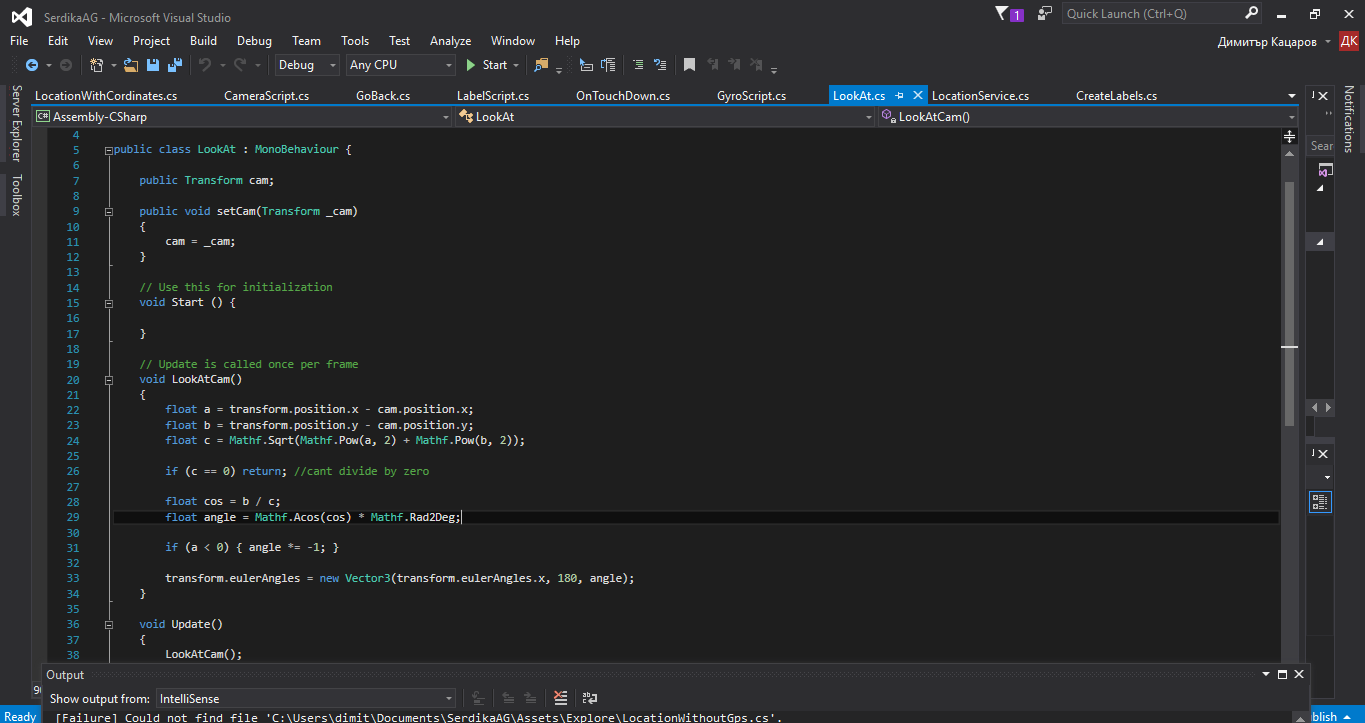
Приравняваме хипотенузата (която служи и за дистанция) на желаната стойност, на която искаме да сложим обекта. Смятат се двете стойности на страните от кординатната система, нужни да получим такава дистанция при съответния синус. Слагаме обекта на тези кординати в Unity.

Смята се дистанцията в метри с метода CalculateDistance. Върната стойност я слагаме на текст седящ под шаблона.



"Фигура 8. LocationWithCoordinates.cs"

LookAt: Скрипта е нужен, за да върти шаблоните така че да са видими изцяло от камерата и да са четими. Използва същия алгоритъм като LocationWithCoordinates, но за да смята под какъв ъгъл трябва да е завъртян обекта,за да гледа към камерата.

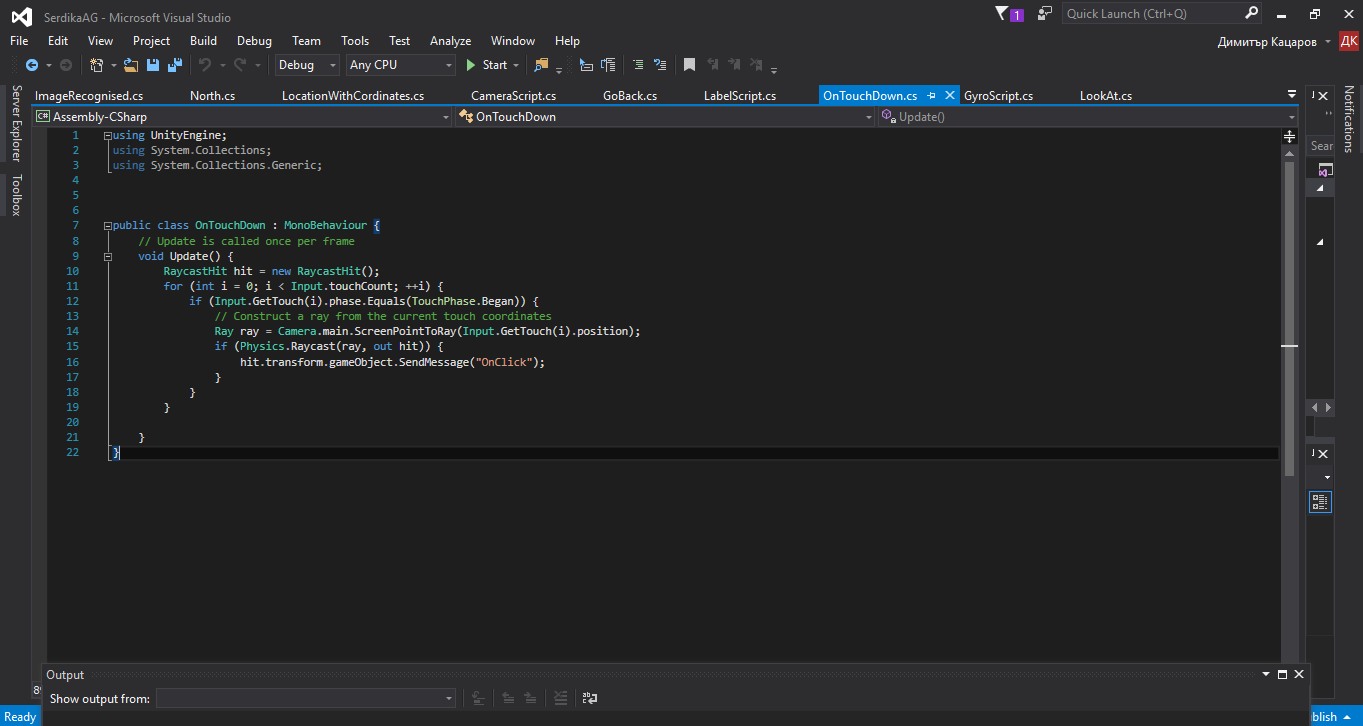


"Фигура 9. LookAt.cs"

**3.1.3. Map**

Сцената зарежда Vuforia. Инициализира камера. При разпознаване на изображение записано в база изтеглена от сайта на Vuforia, се визуализира 3D обект, който е ‘дете‘ на ‘image target’ prefab.

OnTouchDown: При натискане на екрана се пуска “raycast” с кординатите на натиснатото място в Unity мярка. Ако този raycast стигне до обекта се активира event и визуализира label. При повторно натискане, ако raycasta уцели визуализирания label се отваря Information сцената.



**3.1.4. LearnMore**

Използва се Vuforia за разпознаване на QR code и се пуска event, отворя Information сцената.

**3.1.5 Information**

Сцената се състои от няколко секции във формата на меню. Създават се от скрипт и се запълват с информация от базата.

Изпълнява се PHP скрипт, който връща данните от съответния Id на обекта от базата от таблица Information. Данните които получава се слагат на секциите. Взимат се всички изображения със Info\_Id равен на Id на обекта за който зареждаме сцената. Изображенията се изтеглят и се зареждат на секциите си.

Потребителя може да навигира през информацията на обекта, който е заред от базата.

**ГЛАВА ЧЕТВЪРТА**

**РЪКОВОДСТВО НА ПОТРЕБИТЕЛЯ**

**4.1. Изисквания на програмния продукт**

Приложението е предназначено за използване на андроид. За да може потребителя да използва приложението, мобилното му устройство трябва да покрива следните критерии:

1. Телефона да има камера.
2. Физически gyroscope
3. Достъп до интернет
4. Позволена GPS навигацията

**4.2. Инсталация**

Приложението може да бъде инсталирано от APK.

За да може да се инсталира е нужно да се разреши на телефона да

инсталира приложения от източник, различен от „Google Play Store”.

Разрешаването става по следния начин: Отива се в настройките на

телефона, от там се избира опцията „Защита“ и в раздел „Администриране

на устройството“ се слага отметка на „Неизвестни източници“.

Трябва да му бъде позволен интернет достъпа. При опит за инсталация ще излезе съобщение, което пита за разрешение.

**4.3. Използване на приложението**

Навигацията в менютата е лесна. При натискането на бутон от главното менюсе отваря кратко описание на сцената, която се опитвате да отворите и бутон “Open”, който служи за потвърждаване.

**4.3.1 Режим Explore**

В Explore сцената трябва телефона да има включени работещ gyroscope. За да разглеждате забежителностите просто вдигнете телефона си така че да снимате с камерата хоризонта. Въртете се, за да откриете шаблони на забележителностите в София.

**4.3.2 Режим Map**

За Map сцената ви е нужна физическа карта на София. Насочете камерата си към картата и ще се визуализират 3D обекти на забележителностите.

**4.3.3 Режим LearnMore**

LearnMore сцената сканира с камерата на телефона изображение и ви отваря информация свързана с него. За целта просто пуснете сцената и насочете камерата да разчете картината.

**ЗАКЛЮЧЕНИЕ**

В настоящата дипломна работа беше създадено мобилно приложение, с което може да се разглеждат забележителностите на София с помощта на обогатена реалност.

Създаде се следната функционално

* Възможност за разглеждане на виртуални добавени шаблони в реалния свят
* Възможност за разглеждане на физическа карта с 3D визуализирани обекти
* Възможност за разчитане на QR code за бързо достъпване до информацията предоставена от приложението.

Бъдещото развитие на проекта би било разширяването на функционалността на приложението:

* Добавяне на опция за разпознаване на сгради.
* Добавяне на опция за дострояване на руини в стария им облик

ИЗПОЛЗВАНА ЛИТЕРАТУРА

СЪДЪРЖАНИЕ