UNIVERSITY OF THE PELOPONNESE
FACULTY OF ECONOMICS AND TECHNOLOGY
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

Postgraduate Thesis

Geek Treasure Hunt: A mobile game for collecting and visualizing broadband networks metrics via Crowdsourcing.

Koziokas Panagiotis
2022201702006

Supervisor:

Nikolaos D. Tselikas
Associate Professor, University of Peloponnese

Tripoli, September 2021

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Μεταπτυχιακή εργασία

Geek Treasure Hunt: Ένα παιχνίδι για συλλογή και απεικόνιση μετρήσεων ευρυζωνικών δικτύων για κινητά τερματικά μέσω πληθοπορισμού.

Κοζιώκας Παναγιώτης
2022201702006

Επιβλέπων:

Νικόλαος Δ. Τσελίκας
Αναπληρωτής Καθηγητής, Πανεπιστήμιο Πελοποννήσου

Τρίπολη, Σεπτέμβριος 2021

Approved by examination committee 2021.

Nikolaos Tselikas (Associate Professor) George Lepouras (Professor) Ioannis Moscholios (Associate Professor)

.....

Koziokas Panagiotis
Department of Informatics and Telecommunications
University of the Peloponnese

Περίληψη

Η παρούσα διπλωματική εργασία ασχολείται με τη σχεδίαση και ανάπτυξη ενός συστήματος για τη συλλογή μετρήσεων σχετικών με τα ευρυζωνικά δίκτυα και την κατάλληλη οπτικοποίησή τους για εξαγωγή συμπεραμάτων. Το σύστημα αυτό αποτελείται από μία ”εφαρμογή-παιχνίδι” για κινητές συσκευές και μία ιστοσελίδα για τη διαχείριση του περιεχομένου του παιχνιδιού και την επεξεργασία των μετρησεών μας μέσω ενός διαδραστικού ταμπλό οπτικοποίησης γεωγραφικών πληροφοριών.

Η εφαρμογή για τα κινητά είναι ένα παιχνίδι θησαυρού, που βασίζεται στη γεωγραφική τοποθεσία των χρηστών και κάνοντας χρήση τεχνολογιών crowdsourcing καταγράφει και αποθηκεύει μετρήσεις ισχύος του λαμβανόμενου σήματος κινητής τηλεφωνίας και γεωγραφικές συντεταγμένες από την τερματική συσκευή του χρήστη σε ένα κεντρικό σύστημα συλλογής. Το παιχνίδι είναι ένα είδος λογισμικού προγράμματος με στόχο την παροχή ψυχαγωγίας και στη συγκεκριμένη διπλωματική παρουσιάζουμε και αναλύουμε όλα τα βήματα που ακολουθήσαμε κατά τη διάρκεια της ανάπτυξής του βασιζόμενοι σε μία προσέγγιση η οποία ονομάζεται ”Κύκλος ζωής ανάπτυξης παιχνιδιών” (GDLC - Game Development Life Cycle). Το GDLC εστιάζει στις τυπικές αρχές μηχανικής για τη δημιουργία λογισμικού (SDLC - Systems Development Life Cycle) οι οποίες βοηθούν να δημιουργηθεί ένα παιχνίδι.

Η ιστοσελίδα που αναπτύξαμε δίνει τη δυνατότητα στους διαχειριστές του συστήματος να μπορούν να επεξεργαστούν το περιεχόμενο του παιχνιδιού όπως είναι οι διάφορες αποστολές και τα αντικείμενα που θα μπορεί να συλλέξει ο παίκτης. Δίνεται επιπροσθέτως η δυνατότητα μέσω χρήσης κατάλληλων φίλτρων στους διαχειριστές να μπορούν να οπτικοποιήσουν τις συλλεγόμενες μετρήσεις βάσει συγκεκριμένων κριτηρίων για την εξαγωγή χρήσιμων συμπερασμάτων. Η διπλωματική εργασία καταλήγει με τα συμπεράσματα καθώς και με τις προτάσεις μας για μελλοντική επέκταση.

Σκοπός της εν λόγω διπλωματικής εργασίας, κάνοντας χρήση των παραπάνω τεχνολογιών, αποτελεί η συγκέντρωση ικανού αριθμού δειγμάτων - μετρήσεων της ισχύος σήματος, έτσι ώστε να καταστεί δυνατή η απεικόνιση και η εκτίμηση της πραγματικής γεωγραφικής ραδιοκάλυψης των ευρυζωνικών δικτύων επικοινωνιών σε γεωγραφικό περιβάλλον σε πραγματικό χρόνο.

Λέξεις Κλειδιά:

Εφαρμογές για κινητά τερματικά, Ιστοσελίδες, Λογισμικό, Python, Postgres, Google maps, Mobile Lifecycle, Έγγραφο σχεδιασμού παιχνιδιού, RSSI, Latex , Πίνακας Ελέγχου Γεωγραφικής Πληροφορίας.

Abstract

This thesis deals with the design and development of a system for the collection of measurements related to broadband networks and their appropriate visualization via a web based dashboard. This system consists of an application for mobile devices and a website for managing the content of the game and filtering our measurements through an interactive geographical information system (GIS).

The mobile application is a treasure hunting game based on the geographical location of users and by using crowdsourcing technologies records and stores measurements of the received mobile signal and geographical coordinates from the user's end device to a centralized system. The game is a kind of software program aimed at providing entertainment and in this diploma we present and analyze all the steps we followed during its development based on an approach called "Game Development Life Cycle" (GDLC - Game Development Life Cycle). GDLC focuses on standard software development engineering (SDLC) systems that help create a game.

The website we have developed enables system administrators to edit the content of the game such as the various missions and items that the player can collect. In addition, through the use of appropriate filters, administrators are able to visualize the collected metrics based on specific criteria. The thesis concludes with the conclusions as well as with our proposals for future expansion.

The purpose of this thesis, using the above technologies, is to collect a sufficient number of samples - measurements of signal strength, so that it is possible to display and estimate the actual geographical radio coverage of broadband communications networks in real geographical environment in real time.

Keywords:

Mobile apps, Web apps, Software, Python, Postgres, Google maps, Geographical Information Dashboard, Mobile Development Lifecycle, Game Document Design, RSSI, KPI, Latex.

Αφιερώνεται σε όσους με βοήθησαν
σε αυτή την εργασία.

Contents

Chapter 1

Introduction

The mobile sector is a rapidly evolving sector which nowadays is in a stage of evolution as the passage from the fourth to the fifth generation is now a fact. In this great evolution the vision is a unified and operational mobile telephony system which will be able to provide a variety of services to its users. Thus, mobile telephony industry is evolving towards a multimedia oriented model which will be able to provide demanding services such as Mobile TV and Mobile Streaming.

Mobile Network Operators do not have control of the great majority of services used by the end user. This creates a notion in the mind of the end user that the sole responsible for the availability and quality of the experience is the carrier, which in fact has no ability to improve or alter any aspect of the service. One of the biggest challenges MNOs face is to understand and measure the subscriber's network experience.

There are several suggestions for collecting data related to the performance of a network through mobile applications. But most suggested solutions have a UI that need user interaction to collect the data and just shows metrics that ordinary users do not understand their importance.

Target of specific thesis is to collect, present and compare metrics for mobile networks, through a location-based crowd sourcing game, and visualize them on a dynamic GIS dashboard. In the context of the present work, a mobile app was developed for smartphones with operating Android system and a management dashboard using restful application programming interface - API - via python Flask Micro framework that allows two applications to talk to each other .

Furthermore, a theoretical approach is also presented for the principles of different generation of mobile telephony technology. Finally, as a summing-up note, all the collected metrics are web accessible via an interactive dashboard for further analysis and filtering.

- In chapter 2 we analyze more thoroughly the generations of mobile networks.
- In chapter 3 crowd sourcing is been analyzed as concept and some common categories of the technology are presented.
- In 4 chapter we describe all the technologies we are going to use for development purposes.

- Then in chapter 5 we present the different steps followed for the design and implementation of our mobile application based on Mobile Application Development Lifecycle model (MADLC) and Game Document Design (GDD) principles along with our game management system and visualization system.
- In chapter 6 there are two different sections with detailed use cases followed by screenshots aimed at the demonstration of mobile app and administration dashboard.
- Finally in chapter 7 we present our conclusions and our proposals for future development.

Figure 1.1: Networks

Chapter 2

Mobile Networks

2.1 History of Mobile Networks

2.1.1 First Generation Networks - 1G

1G (First Generation) is the name given to the first generation of mobile telephone networks. These systems used analogue circuit-switched technology, with FDMA (Frequency Division Multiple Access), and worked mainly in the 800-900 MHz frequency bands. The networks had a low traffic capacity, unreliable handover, poor voice quality, and poor security.First Generation mobile phone networks were the earliest cellular systems to develop,and they relied on a network of distributed transceivers to communicate with the mobile phones. First Generation phones were also analogue, used for voice calls only, and their signals were transmitted by the method of frequency modulation. These systems typically allocated one 25 MHz frequency band for the signals to be sent from the cell base station to the handset, and a second different 25 MHz band for signals being returned from the handset to the base station. These bands were then split into a number of communications channels, each of which would be used by a particular caller [1].

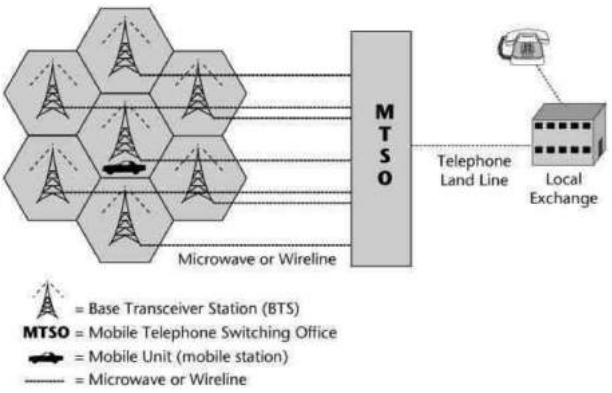


Figure 2.1: Diagram of 1G Network

First-generation networks only allowed voice transmission and, of course, text or data exchange was not supported. In our country (Greece) there has never been a first generation mobile telephony network. The inherent advantages of digital technology over analog meant the immediate replacement of first generation networks by 2G networks.

Also, in the first generation networks there was no security because of it absence of encryption

between conversations. Hence a signal could be easily detected with frequency scanning tools which made the system in terms of security sensitive.

2.1.2 Second Generation Networks - 2G

2G (or 2-G) is short for second-generation cellular network. 2G cellular networks were commercially launched on the GSM standard in Finland by Radiolinja (now part of Elisa Oyj) in 1991. Three primary benefits of 2G networks [3] over their predecessors were:

- Digitally encrypted phone conversations, at least between the mobile phone and the cellular base station but not necessarily in the rest of the network.
- Significantly more efficient use of the radio frequency spectrum enabling more users per frequency band.
- Data services for mobile, starting with SMS text messages.

2G technologies enabled the various networks to provide the services such as text messages, picture messages, and MMS (multimedia messages).

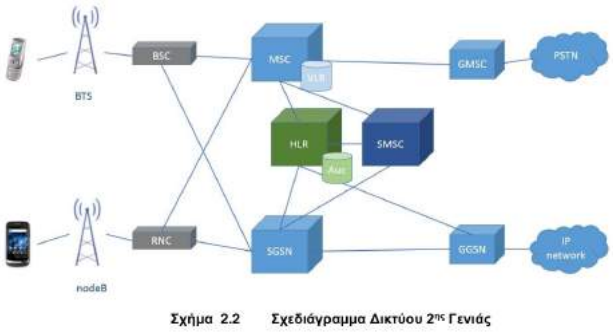


Figure 2.2: Networks

The second generation 2G communications network started in 1991 with data transmission capability at a nominal speed of 9.6 Kbit / s. GSM (whose original name came from Groupe Spécial Mobile, which later renamed Global System for Mobile Communications) is the a global standard that describes digital mobile telephony, secured with an encryption key for a higher degree of security, in relation to the networks of the previous generation.

2.1.3 Third Generation Networks - 3G

3G (short for third generation) is the third generation of wireless mobile telecommunications technology. It is the upgrade for 2.5G GPRS and 2.75G EDGE networks, for faster data transfer. This is based on a set of standards used for mobile devices and mobile telecommunications use services and networks that comply with the International Mobile Telecommunications-2000 (IMT-2000) specifications by the International Telecommunication Union. 3G finds application in wireless voice telephony, mobile Internet access, fixed wireless Internet access, video calls and mobile TV.

3G telecommunication networks support services that provide an information transfer rate of at least 144 kbit/s. Later 3G releases, often denoted 3.5G and 3.75G, also provide mobile broadband

access of several Mbit/s to smartphones and mobile modems in laptop computers. This ensures it can be applied to wireless voice telephony, mobile Internet access, fixed wireless Internet access, video calls and mobile TV technologies [4].

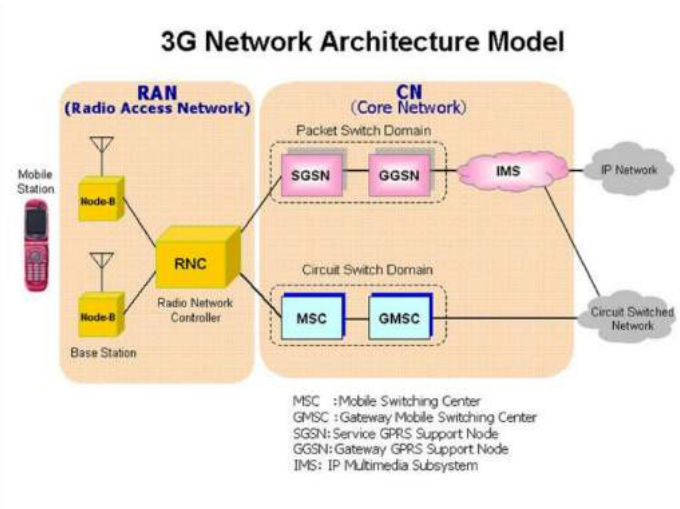


Figure 2.3: Networks

The main goal of the development of third generation mobile networks is providing mobile services ”anywhere” and ”anytime”. This means that a third-generation mobile network user may be mobile wherever it is served even in areas where it does not exist coverage by third generation systems but there are other types of wireless networks.

The ”representative” [2] of the third generation in terms of network development wireless communication is UMTS technology, and it is to this day the most widespread technology in use by wireless mobile networks communications. Provides broadband packet data transmission capabilities at rates of 2 Mbps offering mobile users phones or computers connected to the mobile Internet digitally high quality services.

The term ”3.5 generation” (3.5G or 3G +) refers to the new generation of mobile phones networks which in addition to WCDMA technology have integrated the High Speed Packet Access (HSPA) technology.

2.1.4 Fourth Generation Networks - 4G

4G is the fourth generation of broadband cellular network technology, succeeding 3G, and preceding 5G. A 4G system must provide capabilities defined by ITU in IMT Advanced. Potential and current applications include amended mobile web access, IP telephony, gaming services, high-definition mobile TV, video conferencing, and 3D television [5].

The first-release Long Term Evolution (LTE) standard was commercially deployed in Oslo, Norway, and Stockholm, Sweden in 1998, and has since been deployed throughout most parts of the world. It has, however, been debated whether first-release versions should be considered 4G LTE.

LTE Advanced (Long Term Evolution Advanced) is a candidate for IMT-Advanced standard, formally submitted by the 4GPP organization to ITU-T in the fall 2009, and expected to be released in 2013. The target of 3GPP LTE Advanced is to reach [6] and surpass the ITU requirements. LTE Advanced is essentially an enhancement to LTE. It is not a new technology, but rather an improve-

ment on the existing LTE network. This upgrade path makes it more cost effective for vendors to offer LTE and then upgrade to LTE Advanced which is similar to the upgrade from WCDMA to HSPA. LTE and LTE Advanced will also make use of additional spectrums and multiplexing to allow it to achieve higher data speeds. Coordinated Multi-point Transmission will also allow more system capacity to help handle the enhanced data speeds. Release 10 of LTE is expected to achieve the IMT Advanced speeds. Release 8 currently supports up to 300 Mbit/s of download speeds which is still short of the IMT-Advanced standards.

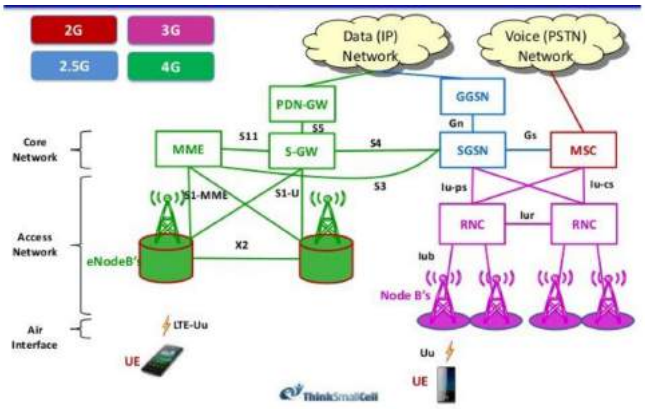


Figure 2.4: Networks

The LTE mobile network, which is now widely known as 4G aims to achieve even higher transmission rates in conjunction with utilizing more bandwidth.

2.1.5 Fifth Generation Networks - 5G

In telecommunications, 5G is the fifth generation technology standard for broadband cellular networks, which cellular phone companies began deploying worldwide in 2019, and is the planned successor to the 4G networks which provide connectivity to most current cellphones. 5G networks are predicted to have more than 1.7 billion subscribers worldwide by 2025, according to the GSM Association. Like its predecessors, 5G networks are cellular networks, in which the service area is divided into small geographical areas called cells. All 5G wireless devices in a cell are connected to the Internet and telephone network by radio waves through a local antenna in the cell. The main advantage of the new networks is that they will have greater bandwidth, giving higher download speeds, eventually up to 10 gigabits per second (Gbit/s). Due to the increased bandwidth, it is expected the networks will not exclusively serve cellphones like existing cellular networks, but also be used as general internet service providers for laptops and desktop computers, competing with existing ISPs such as cable internet, and also will make possible new applications in internet of things (IoT) and machine to machine areas. 4G cellphones are not able to use the new networks, which require 5G enabled wireless devices [7].

5G networks are digital cellular networks, in which the service area covered by providers is divided into small geographical areas called cells. Analog signals representing sounds and images are digitized in the telephone, converted by an analog-to-digital converter and transmitted as a stream of bits. All the 5G wireless devices in a cell communicate by radio waves with a local antenna array and low power automated transceiver (transmitter and receiver) in the cell, over frequency channels assigned by the transceiver from a pool of frequencies that are reused in other

cells. The local antennas are connected with the telephone network and the Internet by a high-bandwidth optical fiber or wireless backhaul connection. As in other cell networks, a mobile device crossing from one cell to another is automatically ”handed off” seamlessly to the new cell. 5G can support up to a million devices per square kilometer, while 4G supports only up to 100,000 devices per square kilometer.The new 5G wireless devices also have 4G LTE capability, as the new networks use 4G for initially establishing the connection with the cell, as well as in locations where 5G access is not available.

Several network operators use millimeter waves for additional capacity, as well as higher throughput. Millimeter waves have a shorter range than microwaves, therefore the cells are limited to a smaller size. Millimeter waves also have more trouble passing through building walls.[10] Millimeter wave antennas are smaller than the large antennas used in previous cellular networks. Some are only a few inches (several centimeters) long.

Massive MIMO (multiple-input multiple-output) was deployed in 4G as early as 2016 and typically used 32 to 128 small antennas at each cell. In the right frequencies and configuration, it can increase performance from 4 to 10 times. Multiple bitstreams of data are transmitted simultaneously. In a technique called beamforming, the base station computer will continuously calculate the best route for radio waves to reach each wireless device and will organize multiple antennas to work together as phased arrays to create beams of millimeter waves to reach the device.

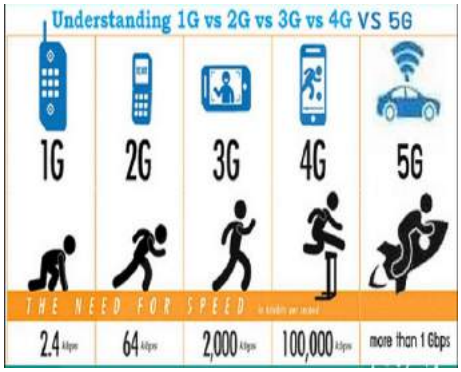


Figure 2.5: Networks

2.2 Mobile Networks Benchmarking

To gain competitive advantage in today’s mobile market, network benchmarking is crucial to Mobile Network Operators (MNOs). Mobile network benchmarking refers to the process of evaluating the MNO’s network performance and comparing the quality of service against competitors. Benchmarking programs enable operators to substantiate market competitiveness, support claims for advertising and marketing campaigns, and optimize their network performance. MNOs use to require independent benchmarking companies to perform drive tests to check coverage and capacity of their networks in order to identify problems and improve network performance in a certain geographical area. The high efforts for executing drive testing results in high costs and therefore in a low frequency of execution, typically this kind of measurement is executed no more than 2-3 times per year in a network while changes in the network and the radio environment happen on a much more frequent basis [8]

Quality of Service (QoS) and Quality of Experience (QoE) are the key factors that drive growth,

increase revenue and reduce churn for network operators and service providers. With the growth in data usage and IoT applications, having a competitive network has become even more vital for subscriber growth. Benchmarking network coverage and data performance against competitors is an efficient way of assessing the areas requiring improvement [9].

For mobile operators, quality of experience (QoE) is one of the key factors that drives differentiation, increases revenue and reduces churn. Nowadays, subscriber growth can only come from competing operators. How good should a mobile network or service be? Benchmarking tests against the competition are an effective way of measuring the marginal improvements that are necessary to trigger focused investments. It is also an effective way of getting to know the competition and to gather valuable information as a source for focused promotional campaigns.

In a mature mobile network, it is fundamental to ensure that the subscriber perceived quality is under control. This means ensuring that QoE keeps improving or is not getting worse, especially when compared to the competition or whenever a new technology is introduced, such as 5G. With the constant demand for new services, better network performance and more capacity, it is crucial that operators have a reliable, accurate and reproducible way to measure their subscribers' QoE. This means comparing themselves with competitors [10].

2.2.1 QoS

The acronym QoS [11] stands for the technical term 'Quality of Service'. QoS defines specific quality requirements which must be met to use a communication service with the desired quality from the subscriber's perspective. In order to meet these requirements, different actions, parameters and processes are defined depending on the network used or the transfer method, summarised under the term QoS. One can essentially distinguish between the transmission quality and the traffic quality. Whilst the traffic quality primarily pertains to the transmission technology and establishing the connection, the transmission quality pertains to the actual information transfer over a network [12].

2.2.2 QoE

Quality of experience (QoE) [13] is a measure of the delight or annoyance of a customer's experiences with a service (e.g., web browsing, phone call, TV broadcast). QoE focuses on the entire service experience it is a holistic concept, similar to the field of user experience [14], but with its roots in telecommunication. QoE is an emerging multidisciplinary field based on social psychology, cognitive science, economics, and engineering science, focused on understanding overall human quality requirements.

2.2.3 KPI

Key performance indicator (KPI) data provide candidate information required for effective network planning, performance analysis and optimization. Key Performance Indicators are metrics put in place and are used to check and grade the quality of service delivered to subscribers on the Network. The efficiency of the obtained reference signal is quite close to the RSRP (reference signal received quality), and the interaction between the two is clear. This is used for handover and re-selection of cells in rankings. RSRP measured in decibel milliwatts (dBm) is the average power taken over resource items in increasing bandwidth that holds reference signals for different cells. The received signal strength indicator (RSSI) is close to that of the RSRP. The distinction is that RSSI is only

determined in OFDM symbols bearing reference symbols from the non-serving co-channel, and serving cells, neighboring channel disturbance, and consumer device thermal noise [15].

It is a fact that mobile network operators (MNOs) need to constantly evolve their networks to support the connectivity demands of the modern societies. As the new technologies (e.g., 4.5, 5G) are adopted, they still need to coexist with current deployments (2G, 3G, 4G), resulting in a dense, complex and heterogeneous network infrastructure. This situation poses new challenges to MNOs in the management of the radio access network, which has operating and maintenance costs implications.

Chapter 3

Crowdsourcing

Crowdsourcing is the practice of engaging a 'crowd' or group for a common goal, often innovation, problem solving, or efficiency. It is powered by new technologies, social media and web 2.0. Crowdsourcing can take place on many different levels and across various industries. Thanks to our growing connectivity, it is now easier than ever for individuals to collectively contribute — whether with ideas, time, expertise, or funds — to a project or cause. This collective mobilization is crowdsourcing [16].

It is a process of tapping in to individuals or groups of people, paid or unpaid who are linked together with a common interest to bring forward powerful, increased results through their aggregated actions or activities.

Figure 3.1: Crowdsourcing

This phenomenon can provide organizations with access to new ideas and solutions, deeper consumer engagement, opportunities for co-creation, optimization of tasks, and reduced costs. The Internet and social media have brought organizations closer to their stakeholders, laying the groundwork for new ways of collaborating and creating value together like never before. The approach is being embraced: "Crowds are a hit. Millions of people, connected by the Internet, are contributing ideas and information to projects big and small. Crowdsourcing, as it is called, is helping to solve tricky problems and providing localized information. And with the right knowledge, contributing to the crowd — and using its wisdom — is easier than ever".

Crowdsourcing involves obtaining work, information, or opinions from a large group of people who submit their data via the Internet, social media, and smartphone apps. People involved in crowdsourcing sometimes work as paid freelancers, while others perform small tasks on a voluntary

basis [17].

3.1 Crowdsourcing types

This section examines in detail a few variations of crowdsourcing, by presenting relevant information:

- Crowdfunding:

  Crowdfunding can be defined as a resource allowing a project initiator to obtain financing from Internet users. This financing can involve all or part of the initiator's capital needs. Even though there is nothing to prevent any project initiator from dedicating a part of his/her website to financing, we have observed that crowdfunding is supported by specialized platforms.

- Crowdvoting:

  Crowdvoting occurs when a website gathers a large group's opinions and judgments on a certain topic. Some of the most famous examples have made use of social media channels: Domino's Pizza, Coca-Cola, Heineken, and Sam Adams have thus crowdsourced a new pizza, bottle design, beer, and song, respectively. In recent years, businesses have been tapping the internet to cheaply and simply survey a large base of potential customers. This "crowdvoting"—a spin on crowdfunding and crowdsourcing—allows companies to ask self-selected website visitors to vote on which prospective new products they would buy [18].

- Crowdcare:

  The concept of "care" is relatively familiar and simple to understand, but it is often bandied about as it is tied up with political concerns. Care is fundamentally a reflection on the role of care for others; a well-meaning empathy combined with concrete assistance. In this context, the crowd, aided by technology, can bring aid to people in difficulty. The members of a crowd may, for example, comfort individuals in distress via material assistance or simply via messages. The outsourced activities are those handled by government services or associations that provide constant assistance to certain categories of the population.

- Crowjobbing:

  Multiple varying routine tasks or a single and very specific task. The principal task must be able to be broken down and then reassembled once the sub-tasks are completed. It must be possible to control the quality of the finished work. The principal task is not directly strategic, though the use of the final result can be of strategic interest. The cost of automating this task must be higher than the cost of outsourcing it.

- Mobile crowdsourcing:

  Mobile crowdsourcing, involves activities that take place on smartphones or mobile platforms that are frequently characterized by GPS technology. This allows for real-time data

gathering and gives projects greater reach and accessibility. However, mobile crowdsourcing can lead to an urban bias, as well as safety and privacy concerns

Active and passive crowdsourcing [19]:

- Passive: Passive Crowdsourcing Description Users/Data sources are sharing information on their own.
- Active: Description Users/Data sources generate information based on requests. Goal To actively design and collect data for a specific task. And then integrate the information.

3.2 Mobile crowdsourcing

Mobile crowdsourcing is a term that describes crowdsourcing activities that are processed on smartphones or other mobile devices. Thanks to the improved, technological smartphone features, including reliable GPS, very good cameras, and continuously new apps, mobile phone users can work on crowdsourcing tasks without any further difficulties. Nowadays, these tasks involve more than simple site descriptions [20].

Figure 3.2: Mobile crowdsourcing

Mobile crowdsourcing can be used to collect data either passively or actively. Users who have smartphones equipped with GPS can be located via apps to create movement profiles. Crowdsourcing is the generalized act of outsourcing tasks, traditionally performed by an employee or contractor, to a large group of Internet population through an open call. With the great development of smartphones with rich built-in sensors and multiple ratio interfaces, mixing smartphone-based mobile technologies and crowdsourcing offers significant flexibility and leads to a new paradigm called mobile crowdsourcing (MCS), which can be fully explored for real-time and location-sensitive crowdsourced tasks [21].

Crowdsourcing using mobile devices, known as mobile crowdsourcing, is a powerful approach incorporating human wisdom into mobile computations to solve problems while exploiting the advantages of mobility and context-awareness. The problems that can be tackled include the use of geographically distributed tasks, and mobile sensing using the collective wisdom of the crowd [22].

In the past decade, crowdsourcing especially mobile crowdsourcing has emerged to facilitate data processing and problem solving. Mobile crowdsourcing refers to a group of people who voluntarily collects and shares data using widely available mobile devices. This data is processed and provided via a data-sharing infrastructure to third parties who are interested in integrating this data. Typically, a mobile crowdsourcing system consists of a platform residing on the cloud and mobile smartphone. Along with the unique multi-sensing capabilities of modern mobile devices, the smart-phone can eventually unfold the potential of crowdsourcing [23].

Mobile crowdsourcing can be utilized for different applications for both scientific and business purposes. Furthermore, Geiger et al. [27] design four basic questions that are important for building the crowdsourcing application: "What is being done?", "Who is doing it?", "Why are they doing it?",and "How is it being done?". From this point, we analyze the mobile crowdsourcing applications along four dimensions: 1) Tasks, 2) Participation, 3) Data Collection, and 4) Processing.

- Task:

  A task in mobile crowdsourcing applications typically refers to a human intelligence required task (which could be aided by machine computations) that employs mobiles and human sensors to collect user data. we can classify the characteristics of crowd tasks into many distinct dimensions as follows.
  Task types:

  Tasks can be divided into two types that could be either human-companion device tasks or human intelligence tasks.The human-companion device tasks utilize mobile devices as sensors to collect human observation and information about environment and infrastructures. Such devices could be smartphones, smart vehicles, wearable devices, and so on that may own several electronic sensors by default (accelerometers, camera, microphone or thermometer)

- Data Contribution:

  - User data : Refers to the extraction of personal contexts(e.g., location, physical activity), health vitals (e.g. heart rate, blood pressure and sugar level) and behavioral patterns (e.g., mobility patterns, daily life patterns)
  - Context data: Regards to information about the surroundings of the users or their status (e.g., noise level of a bus stop, traffic dynamics of a street) or the semantics(the logical type) of a particular area.
  - Social data: Refers to user-generated data in mobile social networks which bridge the gap between online interaction and physical elements (e.g., check-in places.

- Participation:

  - User involvement: Based on the involvement of participants in sensing actions,mobile crowdsourcing can be categorized as: opportunistic and participatory. Opportunistic crowdsourcing is more autonomous, and user involvement is minimal and the mobile device itself makes decisions according to the sensed and stored data e.g., continuous location sampling without the explicit action of the user. Participatory sensing, the active involvement of individuals to contribute sensor data (e.g., taking a picture,reporting a road closure, sentiment analysis) is required.

  - Location-awareness: With the ubiquity of interactive mobile devices providing location awareness and network connectivity, every person with a smartphone can act as a sensor collecting and sharing various types of spatio-temporal data instantaneously. Crowdworkers exploit multi-mobile sensors such as accelerometers, gyroscopes, GPS, cameras and microphones to publish locations, photos, messages and voices via mobile crowd-sourcing applications or social media sites

- Data collection:

  Traditional methods of data management in crowdsourcing generally considered only centralized or client-server communication while mobile ad-hoc network communication involves multi-hop transfers and decentralized processing.A node disseminates tasks among crowd workers through mobile peers in its range, obtains responses from such mobile devices, and integrates the responses to obtain real-time answers.

- Processing:

  In general, mobile crowdsourcing applications exploit wireless sensing networks to sense, transmit, and process data and crowdsourcing tasks. The data of these systems is processed and provided via a data-sharing infrastructure to third parties who are interested in integrating this data. Typically, a mobile crowdsourcing system consists of a platform residing on the cloud and mobile devices.

Chapter 4

Technologies

The main objective of this chapter is to make a short description to all programming languages and demonstrate all the technologies used in the development process of our application and the administration dashboard.

4.1 Android

Android is a mobile operating system based on a modified version of the Linux kernel and other open source software, designed primarily for touchscreen mobile devices such as smartphones and tablets. Android is developed by a consortium of developers known as the Open Handset Alliance and commercially sponsored by Google. It was unveiled in November 2007, with the first commercial Android device launched in September 2008 [24].

Figure 4.1: Android

Android is an operating system designed with mobile in mind, the place where your phone's functions and applications live. Everything you see on the display of your device is a part of the operating system. When you get a call, text message, or email, the OS processes that information and puts it in a readable format [25].

4.1.1 Architecture

Android operating system is a stack of software components which is roughly divided into five sections and four main layers as shown below in the architecture diagram [26].

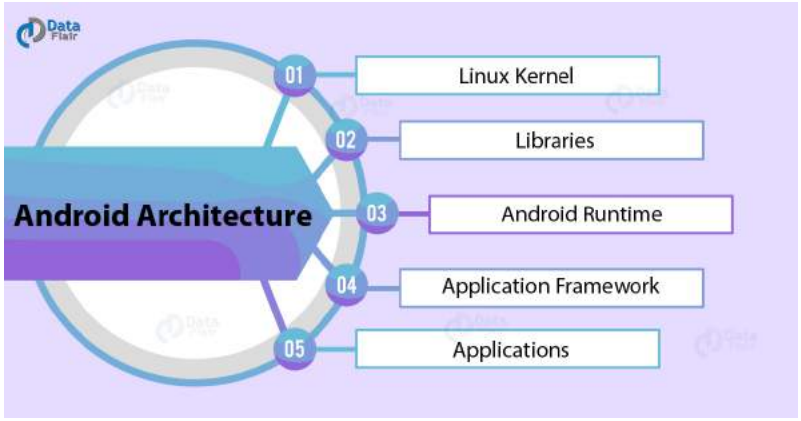


Figure 4.2: Android Architecture

- Linux kernel:

  At the bottom of the layers is Linux - Linux 3.6 with approximately 115 patches. This provides a level of abstraction between the device hardware and it contains all the essential hardware drivers like camera, keypad, display etc.

- Libraries:

  On top of Linux kernel there is a set of libraries including open-source Web browser engine WebKit, well known library libc, SQLite database which is a useful repository for storage and sharing of application data, libraries to play and record audio and video, SSL libraries responsible for Internet security etc.

- Android Runtime:

  This is the third section of the architecture and available on the second layer from the bottom. This section provides a key component called Dalvik Virtual Machine which is a kind of Java Virtual Machine specially designed and optimized for Android. The Dalvik VM enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine.

- Application Framework:

  The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications.

- Applications:

  You will find all the Android application at the top layer. You will write your application to be installed on this layer only. Examples of such applications are Contacts Books, Browser, Games etc.

4.1.2 Activities

An activity represents [27] a single screen with a user interface just like window or frame of Java.Android activity is the subclass of ContextThemeWrapper class. If you have worked with C, C++ or Java programming language then you must have seen that your program starts from main() function. Very similar way, Android system initiates its program with in an Activity starting with a call on onCreate() callback method. There is a sequence of callback methods that start up an activity and a sequence of callback methods that tear down an activity as shown in the below Activity life cycle diagram:

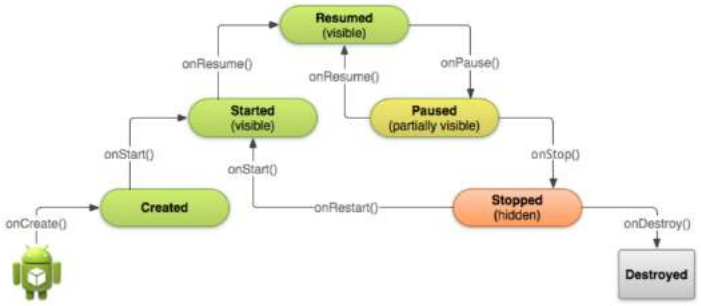


Figure 4.3: Android Lifecycle

4.1.3 Retrofit

Retrofit is type-safe REST client for Android and Java which aims to make it easier to consume RESTful web services [28]. Retrofit is a type-safe REST client for Android, Java and Kotlin developed by Square. The library provides a powerful framework for authenticating and interacting with APIs and sending network requests with OkHttp. See this guide to understand how OkHttp works.

This library makes downloading JSON or XML data from a web API fairly straightforward. Once the data is downloaded then it is parsed into a Plain Old Java Object (POJO) which must be defined for each ”resource” in the response.

4.2 Python

Python is an interpreted [29], object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python’s simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python

Figure 4.4: Python

programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

4.2.1 Flask Microframework

- What is a Web Framework?:

  A Web Application Framework or a simply a Web Framework represents a collection of libraries and modules that enable web application developers to write applications without worrying about low-level details such as protocol, thread management, and so on.

  Flask is a web framework [30], it's a Python module that lets you develop web applications easily. It's has a small and easy-to-extend core: it's a microframework that doesn't include an ORM (Object Relational Manager) or such features.

  It does have many cool features like url routing, template engine. It is a WSGI web app framework.

- What is Flask?:

  Flask is a web application framework written in Python. It was developed by Armin Ronacher [31], who led a team of international Python enthusiasts called Poocco. Flask is based on the Werkzeg WSGI toolkit and the Jinja2 [32] template engine.Both are Pocco projects. WSGI

  The Web Server Gateway Interface (Web Server Gateway Interface, WSGI) has been used as a standard for Python web application development. WSGI [33] is the specification of a common interface between web servers and web applications. Werkzeug

  Werkzeug [34] is a WSGI toolkit that implements requests, response objects, and utility functions. This enables a web frame to be built on it. The Flask framework uses Werkzeug as one of its bases.

  Jinja2 is a popular template engine for Python.A web template system combines a template with a specific data source to render a dynamic web page.

Figure 4.5: Pyhton Flask

4.2.2 Flask Microframework Admin

In a world of micro-services and APIs, Flask-Admin [35] solves the boring problem of building an admin interface on top of an existing data model. With little effort, it lets you manage your web service's data through a user-friendly interface.

The basic concept behind Flask-Admin, is that it lets you build complicated interfaces by grouping individual views together in classes: Each web page you see on the frontend, represents a method on a class that has explicitly been added to the interface.

These view classes are especially helpful when they are tied to particular database models, because they let you group together all of the usual Create, Read, Update, Delete (CRUD) view logic into a single, self-contained class for each of your models.

4.3 Googlemaps

Google Maps is a Web-based service [36] that provides detailed information about geographical regions and sites around the world. In addition to conventional road maps, Google Maps offers aerial and satellite views of many places. In some cities, Google Maps offers street views comprising photographs taken from vehicles.

Figure 4.6: Googlemaps

Google Maps offers several services as part of the larger Web application, as follows:

- A route planner offers directions for drivers, bikers, walkers, and users of public transportation who want to take a trip from one specific location to another.

- The Google Maps application program interface (API) makes it possible for Web site administrators to embed Google Maps into a proprietary site such as a real estate guide or community service page.
- Google Maps for Mobile offers a location service for motorists that utilizes the Global Positioning System (GPS) location of the mobile device (if available) along with data from wireless and cellular networks.
- Google Street View enables users to view and navigate through horizontal and vertical panoramic street level images of various cities around the world.
- Supplemental services offer images of the moon, Mars, and the heavens for hobby astronomers.

4.4 Postgres

PostgreSQL is a powerful, open source object-relational database system that uses and extends the SQL language combined with many features that safely store and scale the most complicated data workloads. The origins of PostgreSQL date back to 1986 as part of the POSTGRES project [37] at the University of California at Berkeley and has more than 30 years of active development on the core platform.

PostgreSQL has earned a strong reputation for its proven architecture, reliability, data integrity, robust feature set, extensibility, and the dedication of the open source community behind the software to consistently deliver performant and innovative solutions. PostgreSQL runs on all major operating systems, has been ACID-compliant since 2001, and has powerful add-ons such as the popular PostGIS [38] geospatial database extender. It is no surprise that PostgreSQL has become the open source relational database of choice for many people and organisations [39].

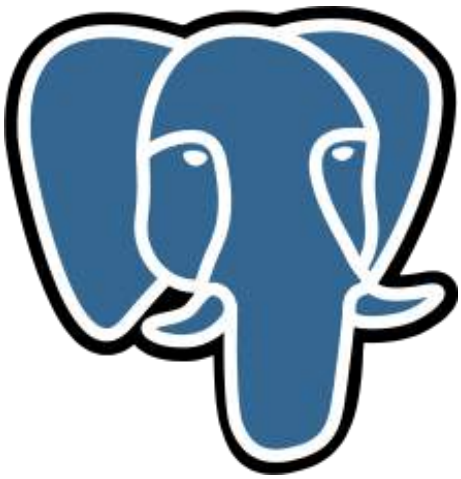


Figure 4.7: Postgres

4.5 User Interface

4.5.1 HTML

HTML is the standard markup language for creating Web pages.

- HTML stands for Hyper Text Markup Language.
- HTML describes the structure of Web pages using markup.
- HTML elements are the building blocks of HTML pages.
- HTML elements are represented by tags.
- HTML elements are the building blocks of HTML pages.
- HTML tags label pieces of content such as ”heading”, ”paragraph”, ”table”, and so on Browsers do not display the HTML tags, but use them to render the content of the page.

Figure 4.8: HTML 5

HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current version of the HTML standard. It was published in October 2014 by the World Wide Web Consortium (W3C) to improve the language with support for the latest multimedia, while keeping it both easily readable by humans and consistently understood by computers and devices such as web browsers.

The most interesting new HTML5 elements are [40]:

- New semantic elements like **<header>**, **<footer>**, **<article>**, and **<section>**.
- New attributes of form elements like number, date, time, calendar, and range.
- New graphic elements:**<svg>** and **<canvas>**.
- New multimedia elements:**<audio>** and **<video>**.

The most interesting new API’s in HTML5 are:

- Geolocation
- Drag and Drop
- Local Storage
- Application Cache
- Web Workers
- SSE

4.5.2 Javascript

Javascript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages.It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as LiveScript, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name LiveScript. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

- JavaScript is a lightweight, interpreted programming language.
- Designed for creating network-centric applications.
- Complementary to and integrated with Java.
- Complementary to and integrated with HTML.
- Open and cross-platform.

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser. It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content. The JavaScript code is executed when the user submits the form, and only if all the entries are valid,they would be submitted to the Web Server.JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly [41].

4.5.3 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content [42].

CSS is created and maintained through a group of people within the W3C called the CSS Working Group. The CSS Working Group creates documents called specifications. When a specification has been discussed and officially ratified by the W3C members, it becomes a recommendation.These ratified specifications are called recommendations because the W3C has no control over the actual implementation of the language.Independent companies and organizations create that software.

NOTE:The World Wide Web Consortiumor W3C is a group that makes recommendations about how the Internet works and how it should evolve.

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a webpage. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, as well as a variety of other effects.

CSS is easy to learn and understand but it provides a powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

4.6 Tools

4.6.1 Swagger Documentation tool

Swagger is an Interface Description Language for describing RESTful APIs expressed using JSON. Swagger is used together with a set of open-source software tools to design, build, document, and use RESTful web services. Swagger includes automated documentation, code generation (into many programming languages), and test-case generation. Swagger's open-source tooling usage can be broken up into different use cases: development, interaction with APIs, and documentation [43].

Figure 4.9: Swagger

Flasgger is a Flask extension to extract OpenAPI-Specification from all Flask views registered in your API. Flasgger also comes with SwaggerUI embedded so you can access `http://localhost:5000/apidocs` and visualize and interact with your API resources.

Flasgger also provides validation of the incoming data, using the same specification it can validates if the data received as as a POST, PUT, PATCH is valid against the schema defined using YAML, Python dictionaries or Marshmallow Schemas.

Flasgger extension

Flasgger [44] is a Flask extension to extract OpenAPI-Specification from all Flask views registered in your API. Flasgger also comes with SwaggerUI embedded so you can access and visualize and interact with your API resources. Flasgger also provides validation of the incoming data, using the same specification it can validates if the data received as as a POST, PUT, PATCH is valid against the schema defined using YAML, Python dictionaries or Marshmallow Schemas.

Flasgger can work with simple function views or MethodViews using docstring as specification, or using swag_from decorator to get specification from YAML or dict and also provides SwaggerView which can use Marshmallow Schemas as specification. Flasgger is compatible with Flask-RESTful so you can use Resources and swag specifications together.

Figure 4.10: Flasgger

4.6.2 Ngrok

What is Ngrok?

Your development machine may be connected to a secure network behind a firewall. To work around access restrictions, ngrok [45] runs a small client process on your machine which creates a private connection tunnel to the cloud service. Your localhost development server is mapped to an ngrok.io sub-domain, which a remote user can then access. There's no need to expose ports, set up forwarding, or make other network changes.

The ngrok client software is available for Windows, macOS, and Linux. Ngrok is a cross-platform application that exposes local server ports to the Internet. Their website claims, so you can spend more time programming—one command for an instant, secure URL to your localhost server through any NAT or firewall."

4.6.3 Postman

Postman is an API client [46] that makes it easy for developers to create, share, test and document APIs. This is done by allowing users to create and save simple and complex HTTP/s requests, as well as read their responses. The result - more efficient and less tedious work.

Figure 4.11: Postman

Postman is a great tool when trying to dissect RESTful APIs made by others or test ones you have made yourself. It offers a sleek user interface with which to make HTML requests, without the hassle of writing a bunch of code just to test an API's functionality.

Chapter 5

Design and Development of the system

The software development methodology (also known as SDM) framework didn't emerge until the 1960s. According to Elliott (2004) [47] the systems development life cycle (SDLC) can be considered to be the oldest formalized methodology framework for building information systems. The main idea of the SDLC has been "to pursue the development of information systems in a very deliberate, structured and methodical way, requiring each stage of the life cycle – from inception of the idea to delivery of the final system – to be carried out rigidly and sequentially" within the context of the framework being applied. The main target of this methodology framework in the 1960's was "to develop large scale functional business systems in an age of large scale business conglomerates. Information systems activities revolved around heavy data processing and number crunching routines".

Figure 5.1: Development

In software engineering, a software development methodology (also known as a system development methodology, software development life cycle, software development process, software process) is a splitting of software development work into distinct phases (or stages) containing activities with the intent of better planning and management. It is often considered a subset of the systems development life cycle. The methodology may include the pre-definition of specific deliverables and artifacts that are created and completed by a project team to develop or maintain an application [48].

The mobile application development life cycle is just a representation of the conventional SDLC from the perspective of mobile. Mobile application development life cycle comprises of

six major stages, which apply to the development of every app [49].

As the mobile applications have complex functionality and are different from the desktop applications, the following Mobile Application Development Life Cycle model (MADLC) is proposed to enable a systematic approach in development.

Phases of MADLC for mobile applications :

- Identification Phase.
- Design Phase.
- Development Phase.
- Prototyping Phase.
- Testing Phase.
- Final Phase.

The game is the one kind of software of program with a goal to provide entertainment. When we plan to start and developing the any games, simply adopting the software development life cycle (SDLC) is not enough for any game developer, as the developer face several challenges during the making any game development life cycle.(challenges like: Graphics, Visuals, Sounds, Animations, Physics, Collisions, AI, Gestures and User inputs, etc) [50].

To overcome the problem of every game developer, a new specific approach will arrive it's known as GDLC(Game Development Life Cycle).

- Identification Phase (Idea Story).

  Here Story/Idea is the actual game concept and requirement of the our project. Those all should mentioned in the requirement/project document. Ideas represents just bits and pieces of your game. It's actual prototype of the game, which have covered all the ideas. The group of ideas are called Game Concept Game Story.

- Game Design Phase (Game Document Design).

  It is a most innovative, creative and complex process of any game. It's must needed to makes a good and quality games. It's requires a critical interactive thinking, understanding, implementing, executing, behavior, UI of the design.

- Technical Requirement Analysis:

  Game Architecture for prototyping of the product written in the Technical Specification Document.

- Development Phase:

  Lead programmer have the major responsibilities about the game development progress and quality, he/she should make the checklist of the pending/working/completed task list as per

the developer's task. Every programmer/developer have to submit their work to lead programmer. Lead developer must follow the code review system as per the game architecture he/she has chosen.

- Testing Phase:

  Testing of the product written in Test Cases and Test Plans.

- Deployment:

  On the publishing platform of the product written as Release Notes Support Document.

Is GDLC as a model to implement exactly inherits SDLC? Software Development product always require a need for an existing problem and its developed for providing a solution, whereas, Game Development product is for entertainment to engage people to have fun, learn and spend good time. Hence the product will require an idea, story, innovation, creativity, writing skills and technical expertise. Hence, just following SDLC engineering principles as a product will have complications and iterations in design and development as a gaming product. Therefore simply adopting the SDLC is not enough, as the developers face several challenges such as (Artwork,Visuals,Animations,Character Interactions,Collisions, Physics, Sounds,etc) during it's lifecycle. To address the problem, we must use a specific approach called GDLC [51].

GDLC focuses on the standard streamlined engineering principles to build a robust software architecture for your game on all the platforms. Computer Science graduates mark this, "software is a need, it is developed to provide "solutions" for a business, but a game, my friend, is never a need for the user instead it feeds the user with fun, creativity, excitement, learning, joy and entertainment".

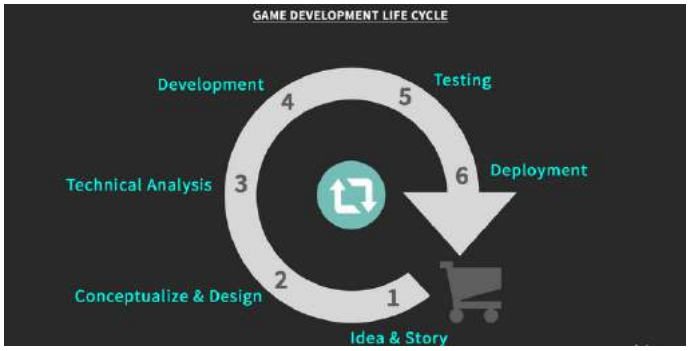


Figure 5.2: Game Development Lifecycle

5.1 Design Phase

In this phase, the idea from the mobile application team is developed into an initial design of the application. The feasibility of developing the application on all mobile platform is determined. Alternatively, the specific target mobile platform is identified. The functional requirements are defined. The software architecture of the application is created. Then the prototypes and associated modules are defined. In the design phase, the requirements will be broken down further to be

able to forecast the project's timeline and estimate the level of effort and amount of resources needed. During this phase, you'll also want to identify any specific designs and workflows for the application.

Our Approach

The requirements analysis considered the first stage in the life cycle of a software product. The requirements analysis is aimed at understanding the needs of users, the conditions in which they work so that the software product to be developed to meet as much as possible their expectations. Eventually a table is presented with this analysis requirements where the functional specifications and the result of the operations displayed.

Table 5.1: Table with Requirements and Specifications for Mobile App

A/A	Specification	Input Data	Necessity level	Result
1	Registration	Username - Password	Necessary	Create a new user
2	First page	First view after login	Necessary	Buttons with app choices
3	Login	Username,Password	Necessary	Login user to app
4	Choose character	Characters view with details	Necessary	Retrieve from database
5	Character view	Character view	Necessary	Storage/Retrieve database
6	Missions view	Details about missions completed	Necessary	Storage/Retrieve database
7	Items view	Details about items collected	Necessary	Storage/Retrieve database
8	Map view	Select/Start/Stop mission	Necessary	Start/Stop collect metrics
9	Google maps view	Specific Id	Necessary	Update Contact
10	Collect item view	Item Id	Necessary	Storage in database

By analyzing the requirements and specifications from this table we decided the mobile app to offer the following:

- Registration Page :
  The user should registered to have access to the app.
- Login Page :
  After registration he can logged in the app with his username and password.
- First page view :
  This is the home screen of the app where the user can see a list of his contacts or products and also can insert them.
- Character views :
  This is view that the player can choose and also see the characteristics of the game characters.
- Missions view :
  The user can explore the completed missions on this view.
- Items view :
  The user can explore the collected items on this view.
- Map view :
  Main view of the game. The player can choose a mission to start exploring and has also the capability to stop the mission.

- Googlemaps view :
  Navigation view through native googlemaps integration.
- Collect item view :
  In this page the user can collect an item when the mission is completed.

We are also braking down the admin views functionalities:

Table 5.2: Table with Requirements and Specifications for Admin Dashboard

A/A	Specification	Input Data	Necessity level	Result
1	Mission view	Fetch missions data	Necessary	Database actions
2	Items views	Fetch items data	Necessary	Database actions
3	Characters view	Fetch characters data	Necessary	Database actions
4	Analytics view	View about metrics	Necessary	Filters view
5	Filters	Filtering options about metrics	Necessary	Filtering Data
6	Google maps view	Plot filtered data	Necessary	Retrieve from database

By analyzing the requirements and specifications from this table we decided the administration dashboard to offer the following:

- Mission view :
  Admin user can perform all CRUD operation related with a mission.
- Items views :
  Admin user can perform all CRUD operation related with items for the game.
- Characters view :
  Admin user can perform all CRUD operation related with items for the game.
- Analytics view :
  Main view for admin users that will be responsible for analyzing the collected metrics.
- Filters :
  Filtering options.
- Google maps view :
  Main component of analytics view for plotting metrics on google maps.
- CRUD operations :
  In computer programming, create, read, update and delete (as an acronym CRUD) are the four basic functions of persistent storage. The acronym CRUD refers to all of the major functions that are implemented in relational database applications. Each letter in the acronym can map to a standard SQL statement. .

  1. Insert Data - **C**reate
  2. Show Data - **R**ead
  3. Update Data - **U**pdate
  4. Delete Data - **D**elete

Requirements - Specifications

5.1.1 Database

The application is required to connect to a database, in which all data will be saved. The role of the database is to provide data to the applications side through interaction between them and also must support multiple functionality.

Database Requirements

The application database must support basic functionality . Below are the basic requirements:

- The database must support storing data in tables and fields, which will used through the application.
- The database should allow the application to execute all CRUD functions.
- The database must support the access of data from multiple users simultaneously.
- The database should support to store sensitive user data, such as coordinates data anonymously

Design of Database

The first step in the database design is to collect all the necessary data to be stored in the database. The Database needs to store data about missions, items and other game functionalities. Also we need to store data for user authentication and all the metrics that are collected. A separate table will be used for geo related data that will not have any relation with the users. It is essentially a representation of our database, and on the basis of this model will do the implementation of the database using SQL. Below is the structure of database schema:

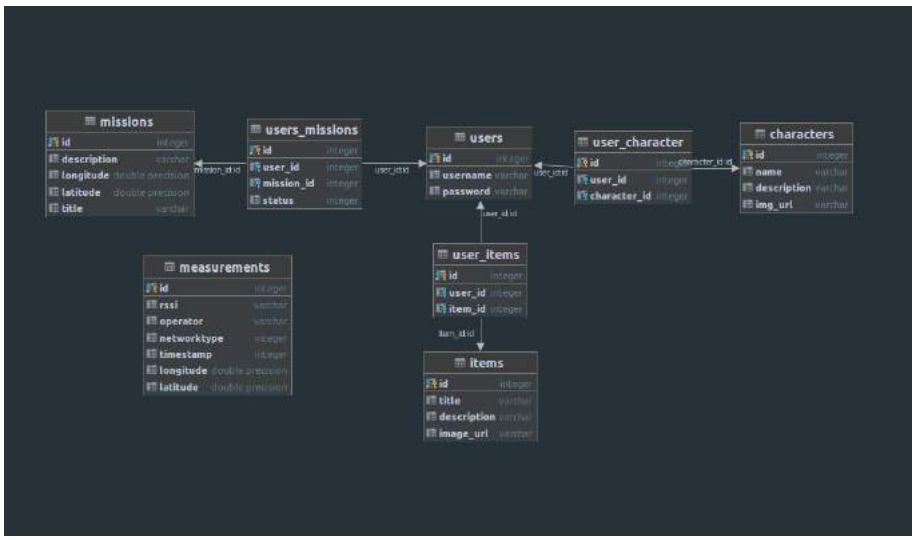


Figure 5.3: Database Schema

5.1.2 Mockups

Pencil is built for the purpose of providing a free and open-source GUI prototyping tool that people can easily install and use to create mockups in popular desktop platforms.

5.2 Game Document Design - GDD

A game design document (often abbreviated GDD) [52] is a highly descriptive living software design document of the design for a video game. A GDD is created and edited by the development team and it is primarily used in the video game industry to organize efforts within a development team. The document is created by the development team as result of collaboration between their designers, artists and programmers as a guiding vision which is used throughout the game development process. When a game is commissioned by a game publisher to the development team, the document must be created by the development team and it is often attached to the agreement between publisher and developer. The developer has to adhere to the GDD during game development process.

5.2.1 Introduction

The GDD serves the purpose of presenting the idea of the game. It is an overview of the game, which can be read by anyone and understand what the game is like.

Game General Idea

”Geek Treasure Hunt” will be a hidden hunting treasure game, which will take place in a real geographical environment as the user will be able to navigate through audio instructions by googlemaps service to search for hidden objects - treasures. The user will be able to select the character he wants and start searching for hidden objects in various places of a map. There will be competition with other players in terms of items they have discovered and missions they have completed.

Game Category

The game could be said to fall into the categories of action and adventure

Audience

The game in its original form will be aimed at users over 12 years old

5.2.2 Game Sequence

Prehistory

After a biblical catastrophe, people lose all access to existing technologies and related objects. Our little explorers (geeks) undertake through the game to discover all the lost objects that are scattered in the world so that the world can regain access to these lost technologies. Each character will have as a subject the exploration of specific technology area.

Characters

Explorers called geeks / nerds each of whom will discover items related to the character description. More specifically, each character will have a specific description, in relation to its subject matter. Depending on his subject, he will discover during the game objects related to his science. Indicatively we will describe 4 characters:

- Computer Science Geek:

  Character who has knowledge of computer science. His main feature is his knowledge of technology.

- Mathematics Geek:

  Character who has knowledge about the science of mathematics.

- Physics Geek:

  Character who has knowledge about the science of physics.

- History Geek:

  Character who has knowledge about the science of history.

Story plot

Starting the game there will be a relevant description which will explain the biblical destruction of our planet. The description will explain to the player that from this disaster many technologies were lost that we had and the only way to access them again is to discover objects, through which we will be able to use them again to help our planet. After the disaster, a group has been formed with the main mission of finding these items. Members of this group are our little explorers, each of whom corresponds to a technology. A brief description of the characters is then made and the player can then choose whoever he wishes.

5.2.3 Basic Characteristics

Playing of the game

The game will be quite easy to play since the user will simply have to follow specific instructions while for the final stage that will have to catch the treasure object, an effort will be made to make the process as easy as possible.

Progress in the game

The player will collect points according to the missions that has successfully completed and the number of items he has collected, and the ranking of the players will be based on these points.

Levels structure

The player at the beginning of each level will be informed about the missions he can choose and the corresponding position of the objects on the map. He will be able to see information about each mission before the selection. The player will then select the mission he/she wants, while at the same time he will receive instructions to go to the desired point.

Level objectives

- Complete mission route.
- Complete mission at a specific time.
- Collect items.
- If the player crosses the route faster than the allotted time he will get extra points.
- Additional points will be awarded for specific geographical points (see chapter ”Hidden game objectives”).

Player movements

The player will move according to the instructions he will receive, within a certain radius based on the point he is in (geo-location.) The player will not have access to the character during the process of navigation to the point where the object is. When he gets there if he has found an item he will be shown a suitable message and will be able to collect the item.

Items - treasures

1. Computer Science Geek:
    - Laptop.
    - Keyboard.
    - Monitor.
    - Mouse.
2. Mathematics Geek:
    - Numbers.
    - Mathematical Symbols.
    - Ruler.
    - Books.
3. Physics Geek:
    - Electricity.
    - Magnet.
    - Atom
4. History Geek:

- Maps.
- Papyrus.

Player actions

- Select the mission he wants to undertake, from a list of missions that will appear on the map.
- For each of the missions, by clicking on the corresponding marker, some information about the specific mission will be displayed, thus giving the possibility to choose based on the description which mission it wishes.
- When the object appears on the screen, he will be able to collect it by pressing specific button.

Plots are basically the descriptions of the missions that the player has to complete. We will present the description of the first two missions of the game as an example to a future expansion of missions.

Plot of first level

Welcome to your first mission!!. Begin your adventure and help the world to overcome the biblic catastrophe. Follow the instructions that will help you collect your first item!!

Plot of second level

You are a great explorer!! After completing your first mission, keep up the good work and continiue exploring your area for more items. Navigate to this locaction to find out if there is hidden treasure waiting for you!!

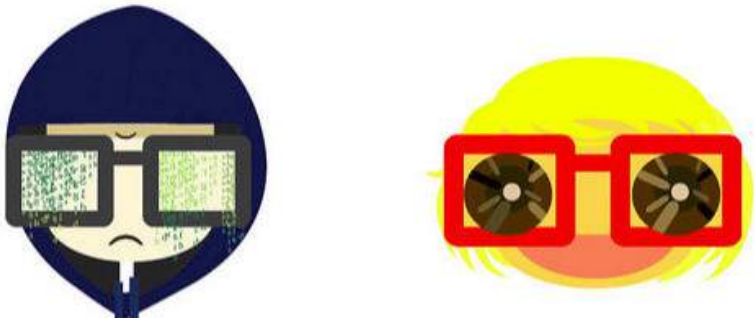
5.2.4 User Interface Mockups

Character icons

Images of available characters:

((a)) First character ((b)) Second Character

Figure 5.4: 2 Figures side by side

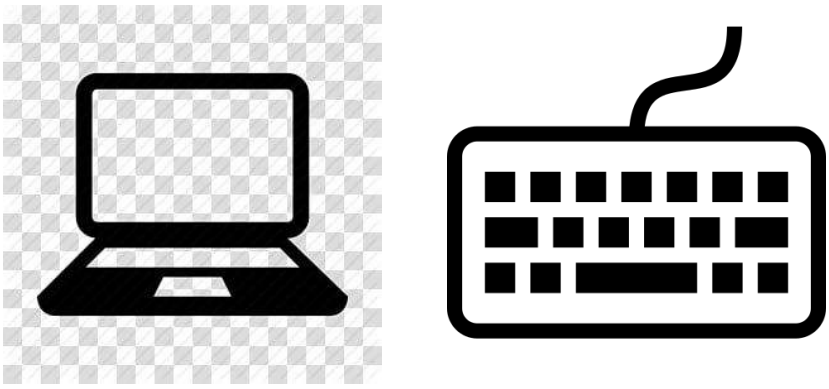


((a)) Third Character ((b)) Fourth Character

Figure 5.5: 2 Figures side by side

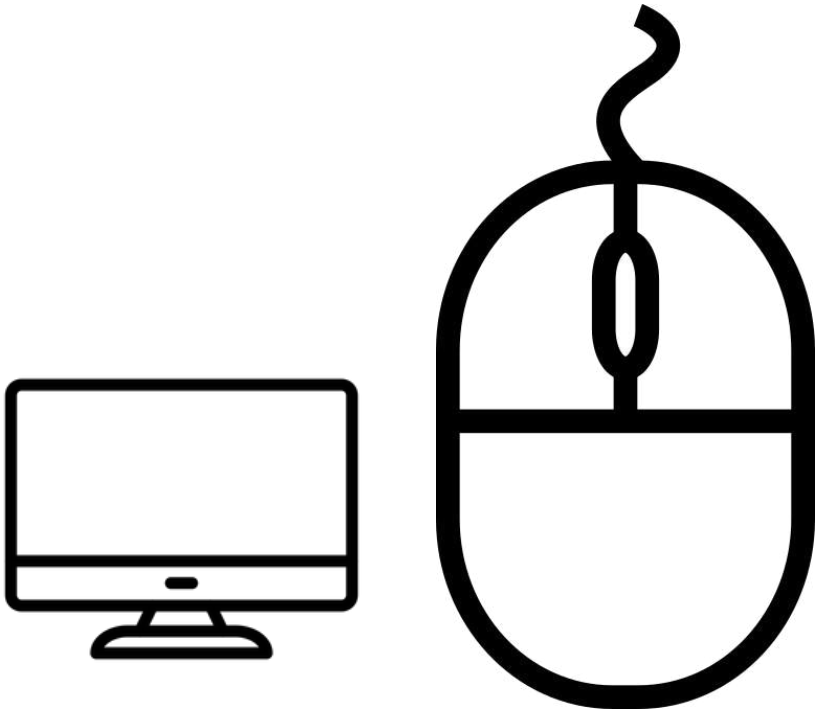
Items

Computer Science Geek items:



((a)) Laptop ((b)) Keyboard

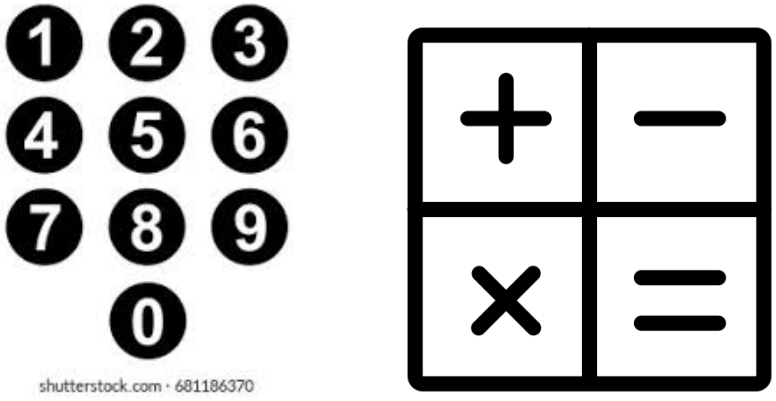
Figure 5.6: Items



((a)) Monitor ((b)) Mouse

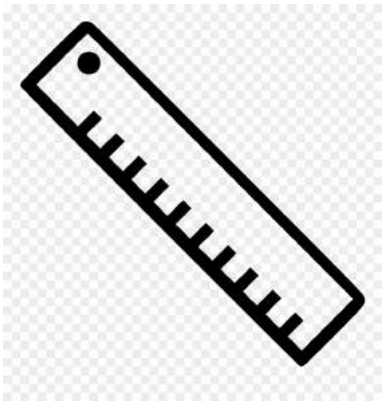
Figure 5.7: Items

Mathematics Science Geek items:

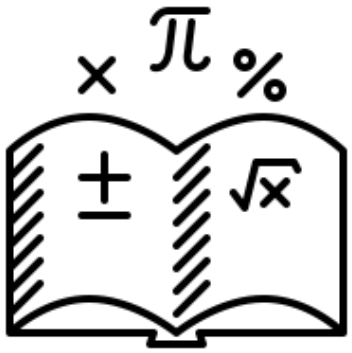


((a)) Numbers ((b)) Math symbols

Figure 5.8: Math Items



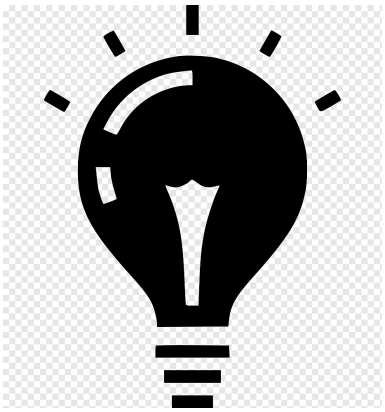
((a)) Ruler



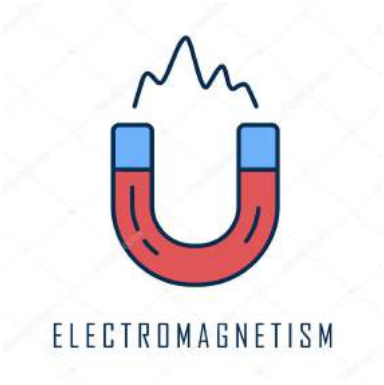
((b)) Math book

Figure 5.9: Math Items

Physic Science Geek items:

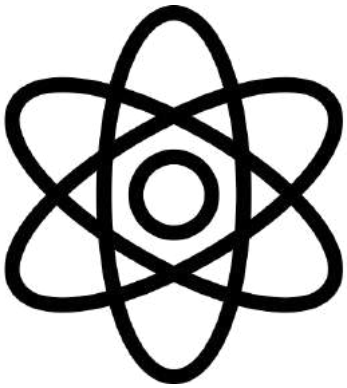


((a)) Electricity



((b)) Magnet

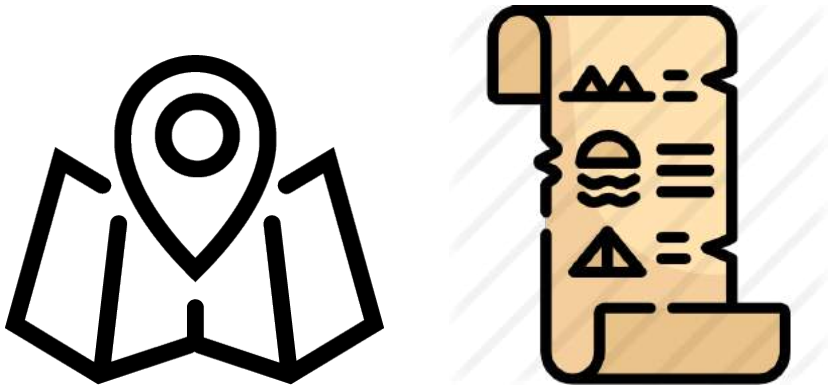
Figure 5.10: Physic items



((a)) Atom

Figure 5.11: Physic item

History Science Geek items:



((a)) History Map ((b)) Papyrus

Figure 5.12: History items

Sounds

The main effect that will give the user a sense of anxiety about the object is looking for will be sonar-based sounds which will become more and more intense as the object approaches. There will also be the following sounds:

- Welcome game sound.
- Sound when the user presses the start button.
- Sound when selecting its mission
- Notification sound that the player has arrived at the item collection point.
- Sound when he manages to pick up the object - treasure

5.3 Development Phase

In this phase, the application is coded. Coding for different modules of the same prototype can proceed in parallel. The development process can be in two stages: Coding for Functional Requirement and Coding for UI requirements. The code is developed first for the core functionalities. Parallel development can be done for modules of the same prototype that are independent of each other. Subsequently, these modules can be integrated.

Our Approach

At this stage we will present the steps followed for the development of our application for our two different modules. At this stage it is essentially planning application gradually trying to come up with a final version. We have three sections - one for the android app, one for the restful API and one for Admin Dashboard - where a variety of screenshots and presentation of some pieces of code will try to demonstrate how we developed our system.

5.3.1 Android Application

Development steps for Android App

Our approach:

For the design and implementation of the mobile application we followed the following steps: we developed a Retrofit client to handle all the request to our Restful API, created all necessary api endpoint in application level and all the activities and services to finalize our app. We continue with some code sample screenshots.

- Mobile app project structure:

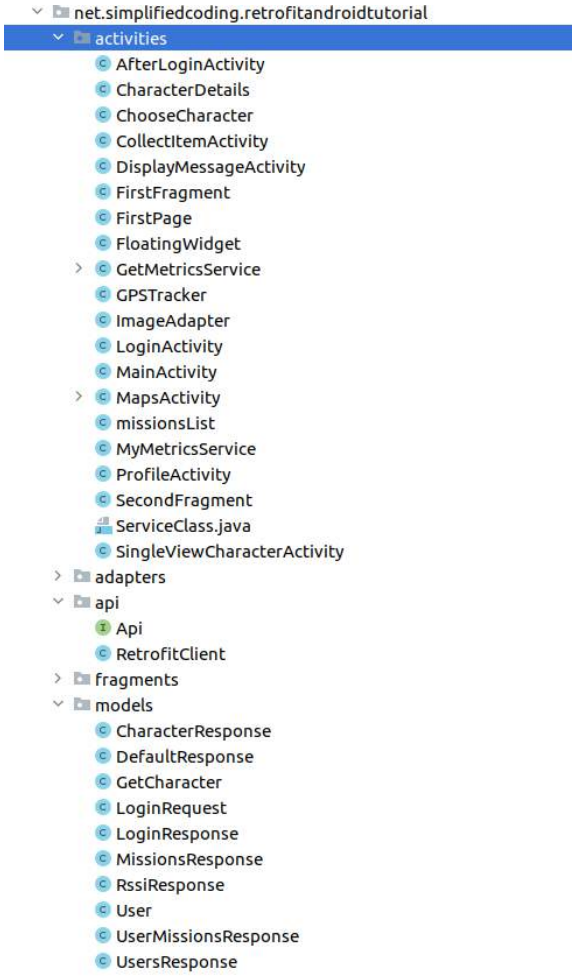


Figure 5.13: Android Structure - Opened

- Retrofit:

  We have a java class which handles all the http requests with Basic Authentication and a base url.

```
import android.util.Base64;
```

```
import java.io.IOException;

import okhttp3.Interceptor;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.Response;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

public class RetrofitClient {

    private static final String AUTH = "Basic " +
        Base64.encodeToString(("pkozi:123456").getBytes(), Base64.NO_WRAP);

    private static final String BASE_URL =
        "http://74293a3bc229.ngrok.io/api/";
    private static RetrofitClient mInstance;
    private Retrofit retrofit;


    private RetrofitClient() {
        OkHttpClient okHttpClient = new OkHttpClient.Builder()
                .addInterceptor(
                        new Interceptor() {
                            @Override
                            public Response intercept(Chain chain) throws
                                IOException {
                            Request original = chain.request();

                            Request.Builder requestBuilder =
                                original.newBuilder()
                                    .addHeader("Authorization", AUTH)
                                    .method(original.method(),
                                        original.body());

                            Request request = requestBuilder.build();
                            return chain.proceed(request);
                        }
                        }
                ).build();

        retrofit = new Retrofit.Builder()
                .baseUrl(BASE_URL)
                .addConverterFactory(GsonConverterFactory.create())
                .client(okHttpClient)
                .build();
    }

    public static synchronized RetrofitClient getInstance() {
        if (mInstance == null) {
```

```
            mInstance = new RetrofitClient();
        }
        return mInstance;
    }

    public Api getApi() {
        return retrofit.create(Api.class);
    }
}
```

- API:

  A screenshot with sample code of our android app java class for API endpoints

```
import
    net.simplifiedcoding.retrofitandroidtutorial.models.CharacterResponse;
import net.simplifiedcoding.retrofitandroidtutorial.models.DefaultResponse;
import net.simplifiedcoding.retrofitandroidtutorial.models.GetCharacter;
import net.simplifiedcoding.retrofitandroidtutorial.models.LoginResponse;
import net.simplifiedcoding.retrofitandroidtutorial.models.MissionsResponse;
import net.simplifiedcoding.retrofitandroidtutorial.models.UsersResponse;
import net.simplifiedcoding.retrofitandroidtutorial.models.RssiResponse;

import retrofit2.Call;
import retrofit2.http.Body;
import retrofit2.http.DELETE;
import retrofit2.http.Field;
import retrofit2.http.FormUrlEncoded;
import retrofit2.http.GET;
import retrofit2.http.Headers;
import retrofit2.http.POST;
import retrofit2.http.PUT;
import retrofit2.http.Path;

public interface Api {


    @FormUrlEncoded
    @POST("register")
    Call<DefaultResponse> createUser(
            @Field("email") String email,
            @Field("password") String password
    );

    @POST("userlogin")
    Call<LoginResponse> userLogin(@Body LoginResponse loginResponse);


    @POST("rssi")
    Call<RssiResponse> sendRssi(@Body RssiResponse rssiResponse);
```

```
    @POST("select_character")
    Call<CharacterResponse> sendCharacter(@Body CharacterResponse
        characterResponse);

    @POST("user_missions")
    Call<MissionsResponse> getMissions(@Body MissionsResponse
        missionsResponse);

    @POST("get_character")
    Call<GetCharacter> getCharacter(@Body GetCharacter getCharacter);

}
```

- GPS integration for geo-location:

  We first listen for location changes while the player is moving via a gps tracker java class we created.

```
package net.simplifiedcoding.retrofitandroidtutorial.activities;

import android.Manifest;
import android.app.AlertDialog;
import android.app.Service;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.IBinder;
import android.provider.Settings;
import android.support.v4.app.ActivityCompat;
import android.util.Log;

public class GPSTracker extends Service implements LocationListener {

    private final Context mContext;

    // flag for GPS status
    boolean isGPSEnabled = false;

    // flag for network status
    boolean isNetworkEnabled = false;

    // flag for GPS status
    boolean canGetLocation = false;

    Location location; // location
    double latitude; // latitude
    double longitude; // longitude
```

```
// The minimum distance to change Updates in meters
private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES = 10; // 10
    meters

// The minimum time between updates in milliseconds
private static final long MIN_TIME_BW_UPDATES = 1000 * 60 * 1; // 1
    minute

// Declaring a Location Manager
protected LocationManager locationManager;

public GPSTracker(Context context) {
    this.mContext = context;
    getLocation();
}

public Location getLocation() {
    try {
        locationManager = (LocationManager)
            mContext.getSystemService(LOCATION_SERVICE);

        // getting GPS status
        isGPSEnabled =
            locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);

        // getting network status
        isNetworkEnabled = locationManager
                .isProviderEnabled(LocationManager.NETWORK_PROVIDER);

        if (!isGPSEnabled && !isNetworkEnabled) {
            // no network provider is enabled
        } else {
            this.canGetLocation = true;
            // First get location from Network Provider
            if (isNetworkEnabled) {
                if (ActivityCompat.checkSelfPermission(this,
                    Manifest.permission.ACCESS_FINE_LOCATION) !=
                    PackageManager.PERMISSION_GRANTED &&
                    ActivityCompat.checkSelfPermission(this,
                    Manifest.permission.ACCESS_COARSE_LOCATION) !=
                    PackageManager.PERMISSION_GRANTED) {
                    // TODO: Consider calling
                    //    ActivityCompat#requestPermissions

                    // here to request the missing permissions, and then
                        overriding
                    //   public void onRequestPermissionsResult(int
                        requestCode, String[] permissions,
                    //                                        int[]
                        grantResults)
```

```
                    // to handle the case where the user grants the
                        permission. See the documentation
                    // for ActivityCompat#requestPermissions for more
                        details.

                    return location;
                }
                locationManager.requestLocationUpdates(
                        LocationManager.NETWORK_PROVIDER,
                        MIN_TIME_BW_UPDATES,
                        MIN_DISTANCE_CHANGE_FOR_UPDATES, this);

                Log.d("Network", "Network");
                if (locationManager != null) {
                    location = locationManager
                            .getLastKnownLocation(LocationManager.NETWORK_PROVIDER);

                    if (location != null) {
                        latitude = location.getLatitude();
                        longitude = location.getLongitude();
                    }
                }
            }

            // if GPS Enabled get lat/long using GPS Services
            if (isGPSEnabled) {
                if (location == null) {
                    locationManager.requestLocationUpdates(
                            LocationManager.GPS_PROVIDER,
                            MIN_TIME_BW_UPDATES,
                            MIN_DISTANCE_CHANGE_FOR_UPDATES, this);

                    Log.d("GPS Enabled", "GPS Enabled");
                    if (locationManager != null) {
                        location = locationManager
                                .getLastKnownLocation(LocationManager.GPS_PROVIDER);

                        if (location != null) {
                            latitude = location.getLatitude();
                            longitude = location.getLongitude();
                        }
                    }
                }
            }
        }

    } catch (Exception e) {
        e.printStackTrace();
    }

    return location;
```

```
}

/**
 * Stop using GPS listener
 * Calling this function will stop using GPS in your app
 * */

public void stopUsingGPS(){
    if(locationManager != null){
        locationManager.removeUpdates(GPSTracker.this);
    }
}

/**
 * Function to get latitude
 * */

public double getLatitude(){
    if(location != null){
        latitude = location.getLatitude();
    }

    // return latitude
    return latitude;
}

/**
 * Function to get longitude
 * */

public double getLongitude(){
    if(location != null){
        longitude = location.getLongitude();
    }

    // return longitude
    return longitude;
}

/**
 * Function to check GPS/wifi enabled
 * @return boolean
 * */

public boolean canGetLocation() {
    return this.canGetLocation;
}

/**
 * Function to show settings alert dialog
 * On pressing Settings button will lauch Settings Options
```

```
 * */

public void showSettingsAlert(){
    AlertDialog.Builder alertDialog = new AlertDialog.Builder(mContext);

    // Setting Dialog Title
    alertDialog.setTitle("GPS is settings");

    // Setting Dialog Message
    alertDialog.setMessage("GPS is not enabled. Do you want to go to
        settings menu?");

    // On pressing Settings button
    alertDialog.setPositiveButton("Settings", new
        DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog,int which) {
            Intent intent = new
                Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
            mContext.startActivity(intent);
        }
    });

    // on pressing cancel button
    alertDialog.setNegativeButton("Cancel", new
        DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int which) {
            dialog.cancel();
        }
    });

    // Showing Alert Message
    alertDialog.show();
}

@Override
public void onLocationChanged(Location location) {
}

@Override
public void onProviderDisabled(String provider) {
}

@Override
public void onProviderEnabled(String provider) {
}

@Override
public void onStatusChanged(String provider, int status, Bundle extras)
    {
}
```

```
    @Override
    public IBinder onBind(Intent arg0) {
        return null;
    }
}
```

- How we retrieve RSSI:

  We implement a service which extends android.telephony.PhoneStateListener and listens to signal strength changes. Based on the network type we collect the signal strength value and sore it to our database. Screenshot with sample code:

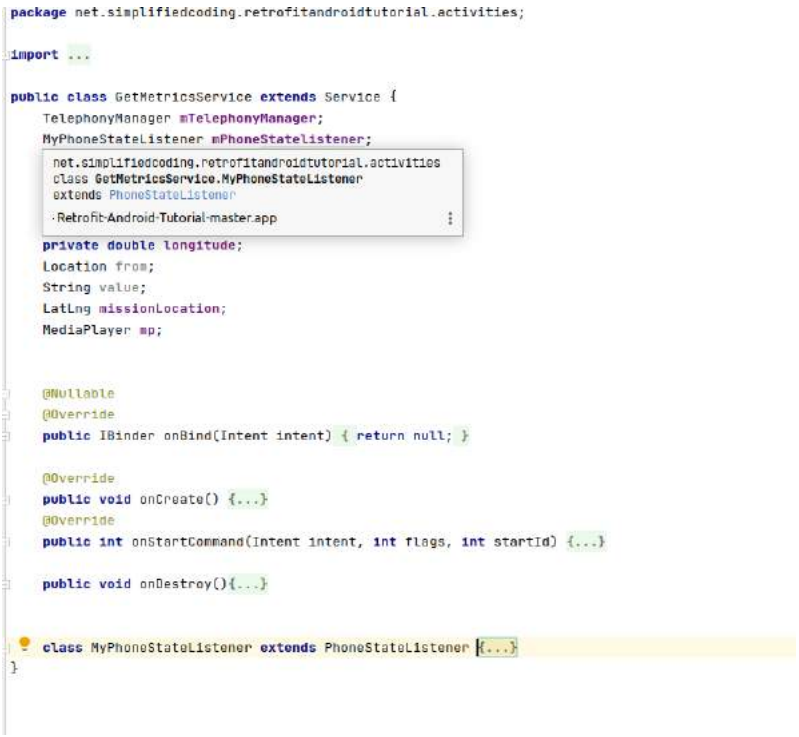


Figure 5.14: Get RSSI

Sample code for getting RSSI value for 4G network category based on getLteRsrp value provided by android system.

```
        else if (network_category == "4G"){
            Toast.makeText(getBaseContext(), "Network is 4G ",
                Toast.LENGTH_LONG).show();

            Method[] methods = SignalStrength.class
                    .getMethods();
            for (Method mthd : methods) {
                if (mthd.getName().equals("getLteRsrp")) {
                    try {
                        rssi =
                            Integer.parseInt(String.valueOf(mthd.invoke(signalStrength)));
                    } catch (IllegalAccessException e) {
                        e.printStackTrace();
```

```
                    } catch (InvocationTargetException e) {
                        e.printStackTrace();
                    }
                }
            }
        }
```

- Googlemaps API:

  Get an API key :

  - Go to the Google API Console
  - Create or select a project
  - Click Continue to enable the API and any related services
  - On the Credentials page, get an API key (and set the API key restrictions)
  - Replace the value of the key parameter in the URL with your own API key (that's the API key that you've just obtained)

- Ngrok.

  Ngrok allows you to expose a web server running on your local machine to the internet. Just tell ngrok what port your web server is listening on. When you start ngrok, it will display a UI in your terminal with the public URL of your tunnel and other status and metrics information about connections made over your tunnel. Ngork helped us develop our app locally without the need to host our api.

  command : ngrok http 5000

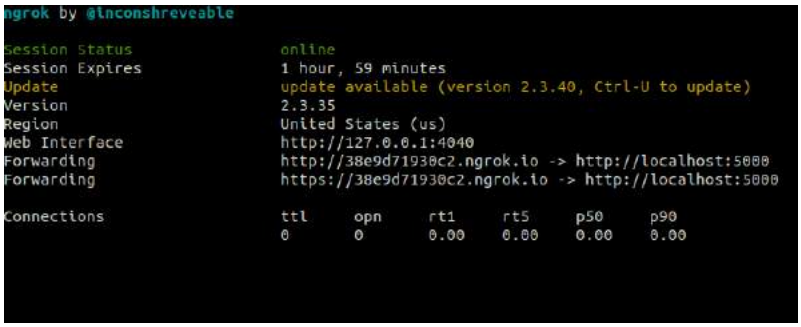


Figure 5.15: Ngrok

  https://ngrok.com/docs

5.3.2 Restful API

Our approach:

For the design and implementation of the API we followed the following steps. Based on the MVC model we created a structure for the project which is presented below and which is fully documented through the extension of flasgger. Also made use of SQLAlchemy which is the Python SQL toolkit and Object Relational Mapper that gives application developers the full power and

flexibility of SQL.

Development steps for Restful API:

- Restful API MVC.

  Model–View–Controller (MVC) is an architectural pattern for implementing user interfaces. It divides an application into three interconnected parts: the Model, the View, and the Controller. Separating the ”internal representations of information from the ways that information is presented to and accepted from the user” allows us to increase modularity for simultaneous development and code reuse. The MVC model or ”pattern” is commonly used for developing modern user interfaces. It is provides the fundamental pieces for designing a programs for desktop or mobile, as well as web applications. It works well with object-oriented programming, since the different models, views, and controllers can be treated as objects and reused within an application. The model stores data that is retrieved according to commands from the controller.

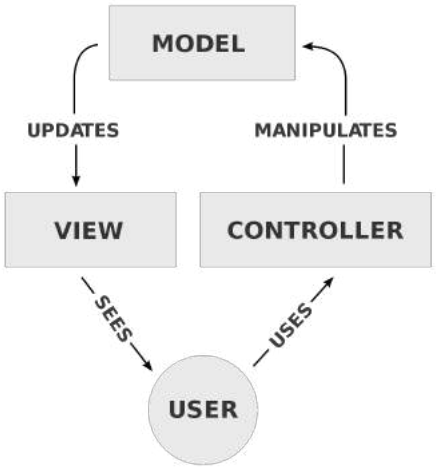


Figure 5.16: MVC

The view generates output for the user based on changes in the model. The controller acts on both model and view, it sends commands to the model to update its state and to the view to change information presented to users.

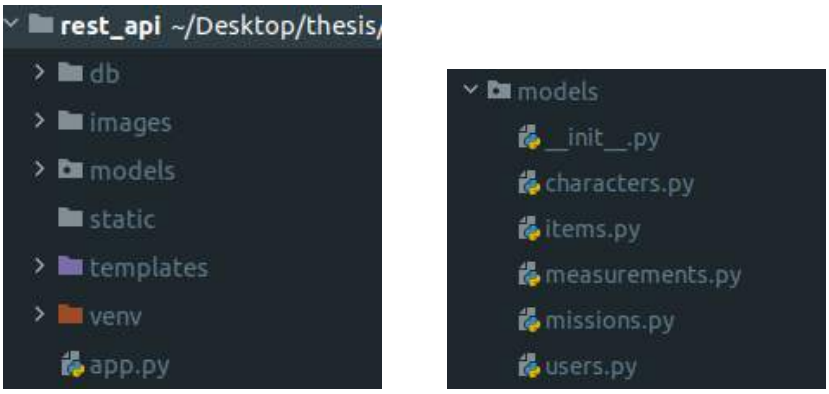
Table 5.3: MVC Components

A/A	Component	Description
1	Model	Handles application data. central component of MVC.
2	View	Renders data from models to UI
3	Controller	Accepts input and converts to commands for model/view

Project Structure:

The structure of our project is the following: An app.py python file which basically has all the code related to the endpoints of our Restful API , models directory which has all code

for handling mapping of our database tables , static directory for static content that needs to be served and finally templates directory for admin/'s dashboard templates



((a)) Project Structure ((b)) Models Structure

Figure 5.17: Structure

```
@app.route('/api/register', methods=['POST'])
def register():...


@app.route('/api/userlogin', methods=['POST'])
def login():...


@app.route('/api/map_measurements', methods=['POST'])
def map_measurements():...


@app.route('/api/map_measurements_by_date', methods=['POST'])
def map_measurements_by_date():...


@app.route('/api/rssi', methods=['POST'])
def rssi():...


@app.route('/api/missions', methods=['GET'])
def missions():...


@app.route('/api/user_missions/', methods=['POST'])
def user_missions():
```

Figure 5.18: Structure API Endpoints

Database model sample python code:

```
from app import db


class Measurements(db.Model):
```

```
    id = db.Column(db.Integer, primary_key=True)
    rssi = db.Column(db.String(80), unique=True, nullable=False)
    operator = db.Column(db.String(80), unique=True, nullable=False)
    networktype = db.Column(db.Integer, unique=True)
    timestamp = db.Column(db.Integer, unique=True)
    longitude = db.Column(db.Float(120), unique=True)
    latitude = db.Column(db.Float(120), unique=True)

    @property
    def serialized(self):
        """Return object data in serializable format"""
        return {
            'id': self.id,
            'rssi': self.rssi,
            'operator': self.operator,
            'networktype': self.networktype,
            'timestamp': self.timestamp,
            'longitude': self.longitude,
            'latitude': self.latitude
        }
```

- Swagger docs:

  We created documentation via flasgger extension for swagger. All API endpoints are fully documented and accessible on http://127.0.0.1:5000/doc url as shown at the screenshot below. As a result, it can be used to share documentation among product managers, testers and developers, but can also be used by various tools to automate API-related processes.

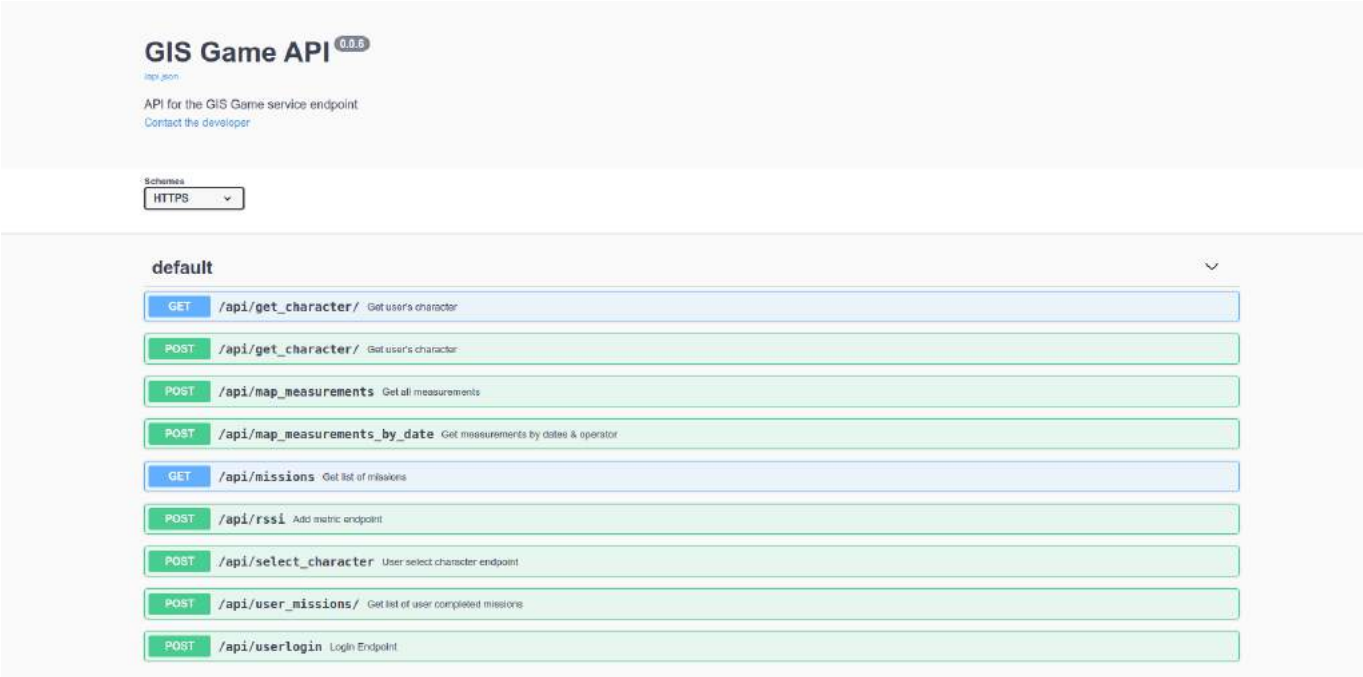


Figure 5.19: Swagger UI

In the example above we describe a simple “Hello World API”. It has a single URL exposed

- "/hello/user". "user" is the single parameter to the API, defined as part of the path, and we say it's a string. We also describe the successful response, and mention it's a string as well. A generic 400 error can be received if the "user" contained invalid characters. You can also see that throughout the operation itself we provide additional documentation.

```
@app.route('/api/userlogin', methods=['POST'])
def login():
    """
     Login Endpoint
    ---
    parameters:
      - name: body
        in: body
        required: true
        schema:
          id: Login
          required:
            - name
          properties:
            username:
              type: string
              description: The product's name.
            password:
              type: string
              description: The product's name.


    responses:
      200:
        description: Success login
      400:
        description: Bad request
```

Figure 5.20: Swagger Login Code

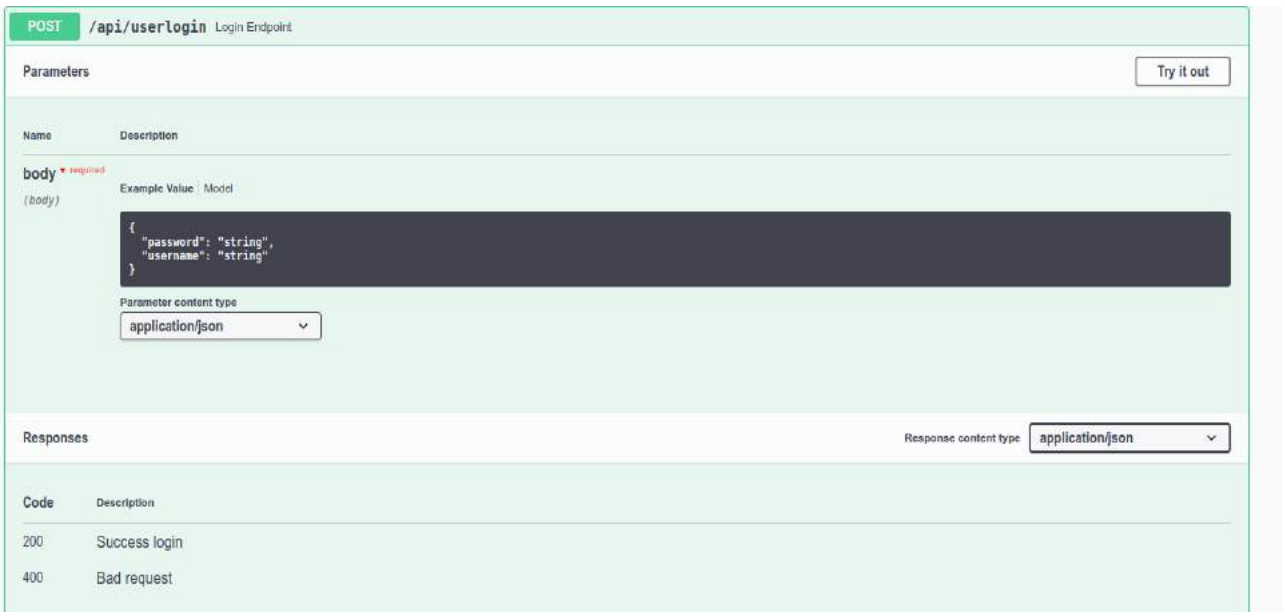


Figure 5.21: Swagger Login UI

- SQLalchemy - ORM:

  SQLAlchemy is most famous for its object-relational mapper (ORM), an optional component that provides the data mapper pattern, where classes can be mapped to the database in open ended, multiple ways - allowing the object model and database schema to develop in a cleanly decoupled way from the beginning. Example of query between ORM and native SQL query is shown below.

```
user_data = Users.query.filter_by(username=username,
                                  password=password).first()
```

Figure 5.22: ORM SQL

```
query_string = """SELECT * FROM users WHERE username=username
                                  AND password=password"""
```

Figure 5.23: Native SQL

5.3.3 CMS Admin Dashboard

Management & Visualization Dashboard

In this subsection we will present two additional components of the system that have as main goal the content management of the game aspects and the visualization of our measurements on a GIS custom solution. A content management system (CMS) is a computer software used to manage the creation and modification of digital content. In our case Flask microframework that comes with a built-in admin area will help us build a custom CMS with the desired views. We will also make an analysis of what method we followed for the visualization on the map.

What is a data visualization dashboard?

A dashboard is a data visualization tool that tracks, analyzes, and displays KPIs, metrics, and critical data points. Dashboards empower both technical and non-technical users to understand and leverage business intelligence to make more informed decisions. Users actively participate in the analytics process by compiling data and visualizing trends or occurrences, and uncovering an objective view of performance metrics that can be immediately understood [53].

A data dashboard is an information management tool that visually tracks, analyzes and displays key performance indicators (KPI), metrics and key data points to monitor the health of a business, department or specific process. They are customizable to meet the specific needs of a department and company. Behind the scenes, a dashboard connects to your files, attachments, services and API's, but on the surface displays all this data in the form of tables, line charts, bar charts and gauges. A data dashboard is the most efficient way to track multiple data sources because it provides a central location for businesses to monitor and analyze performance. Real-time monitoring reduces the hours of analyzing and long line of communication that previously challenged businesses.

Development steps for Admin Dashboard

Our approach:

Our idea is to give the capability to the admin users of the system to manage various game related functionalities. The admin will have the following capabilities:

- Add/Edit/Delete functionality for a mission with title, description and coordinates fields.
- Add/Edit/Delete functionality for items with image , title and description fields.
- Add/Edit/Delete functionality for characters with image , title and description fields.
- GIS (Geographical Information System) Dashboard to visualize collected metrics.
- Admin Structure:

  Model views allow you to add a dedicated set of admin pages for managing any model in your database. Do this by creating instances of the ModelView class, which you can import from one of Flask-Admin's built-in ORM backends . Example of how we can add views to admin dashboard is presented below with code sample .

```
class MissionsView(ModelView):
    edit_template = 'edit_mission.html'
    create_template = 'create_mission.html'


class AnalyticsView(BaseView):
    @expose('/')
    def index(self):

        hello = "hello"
        return self.render('missions.html', hello=hello)


class UserModelView(ModelView):
    def is_accessible(self):
        return False


class CharactersModelView(ModelView):
    def is_accessible(self):
        return True


admin = Admin(app, name='Dashboard', template_mode='bootstrap3')

admin.add_view(ModelView(Items, db.session))
admin.add_view(MissionsView(Missions, db.session))
admin.add_view(AnalyticsView(name='Analytics', endpoint='analytics'))
```

```
admin.add_view(UserModelView(Users, db.session))
```

Straight out of the box, this gives you a set of fully featured CRUD views for your model. There are many options available for customizing the display and functionality of these built-in views.

- UI (User Interface).

  In templates directory 5.24 we have all the .html files for user interface components of admin dashboard. There is a main index html file for base code and the other files are used to extend admin views functionalities.

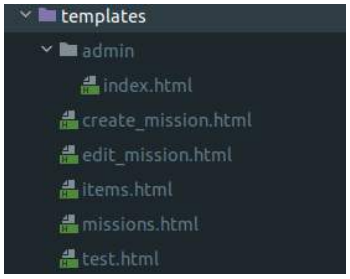


Figure 5.24: Templates Structure

An admin user can navigate to to view CMS content. Information about functionalities of views is been displayed. There are three tabs that admin can use to edit the content of the game and one tab for filter and visualize collected metrics.

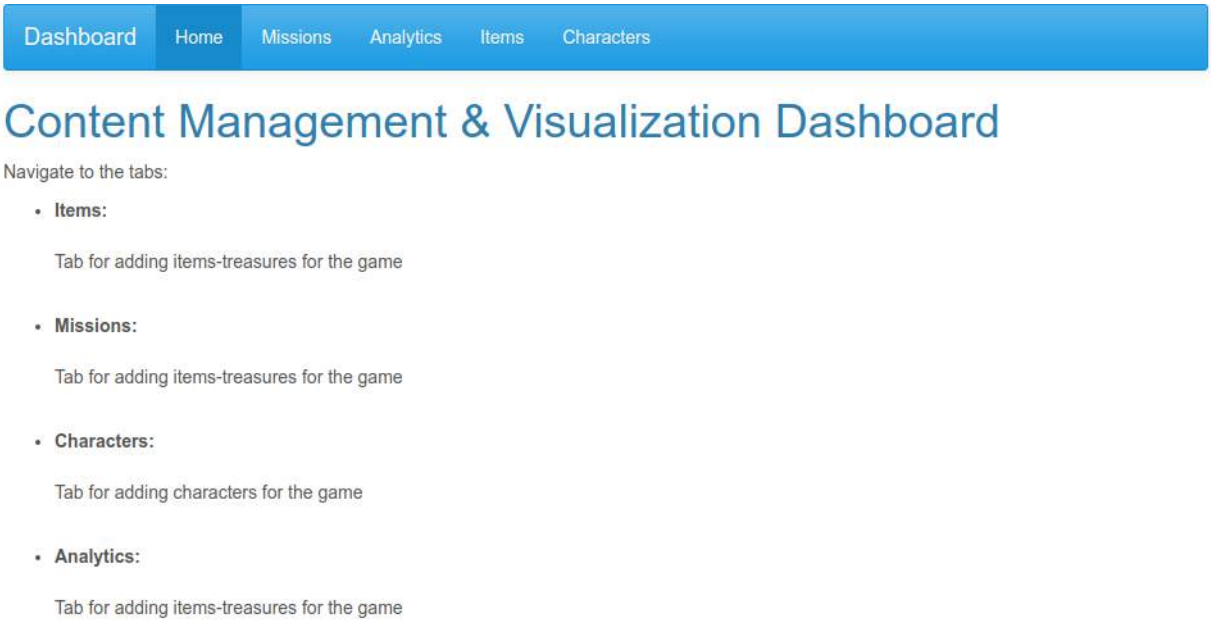


Figure 5.25: CMS Information

Admin can create a mission with all the required fields that exposed directly from mission table of our database schema.

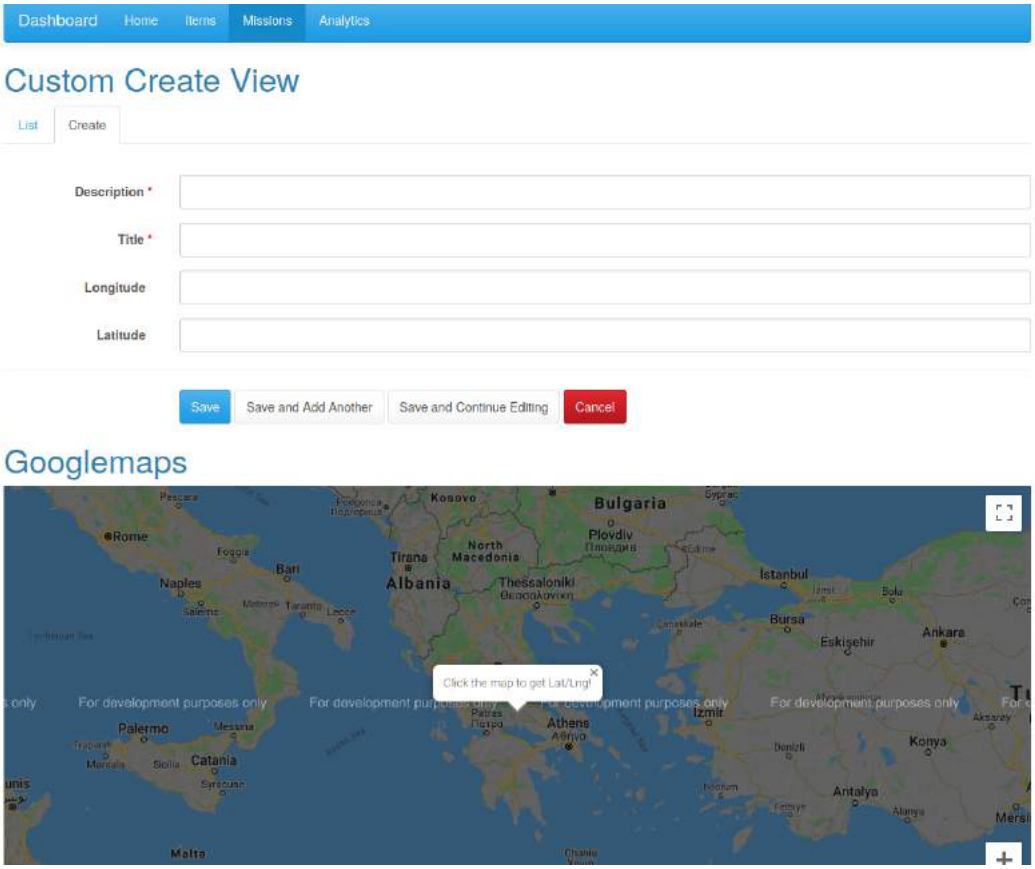


Figure 5.26: Create new mission

He can view a list of all missions and also update or delete an entry from database and can view a list of all items - treasures and also update or delete an entry from database.

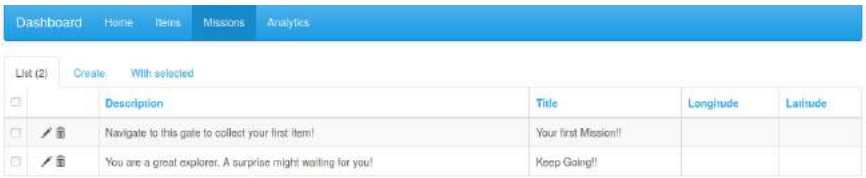


Figure 5.27: Lists Example

It is worth mentioning that all changes or updates to the content from the admin area via the api become available immediately to the application layer. Below are some screenshots from the admin area regarding the mentioned functionality.

Visualization System

The visualization dashboard is basically a custom GIS system that will have the functionality to plot all the collected metrics that will help us summarize to some conclusions about the quality of mobile networks. To visualize our measurements we designed and created a gis dashboard which, with the use of appropriate filters will help in their further analysis. The filters we implemented are the following:

- Metric KPI:
  RSSI
- Provider:
  Broadband Network Providers
- Network Type:
  2G, 3G, 4G and 5G
- Start/End Datetime field:
  Date time fields for filtering

Below are some screenshots from the admin area regarding the mentioned functions. We can see an overview of the dashboard with all the available filters and a map which is used for their visualization.

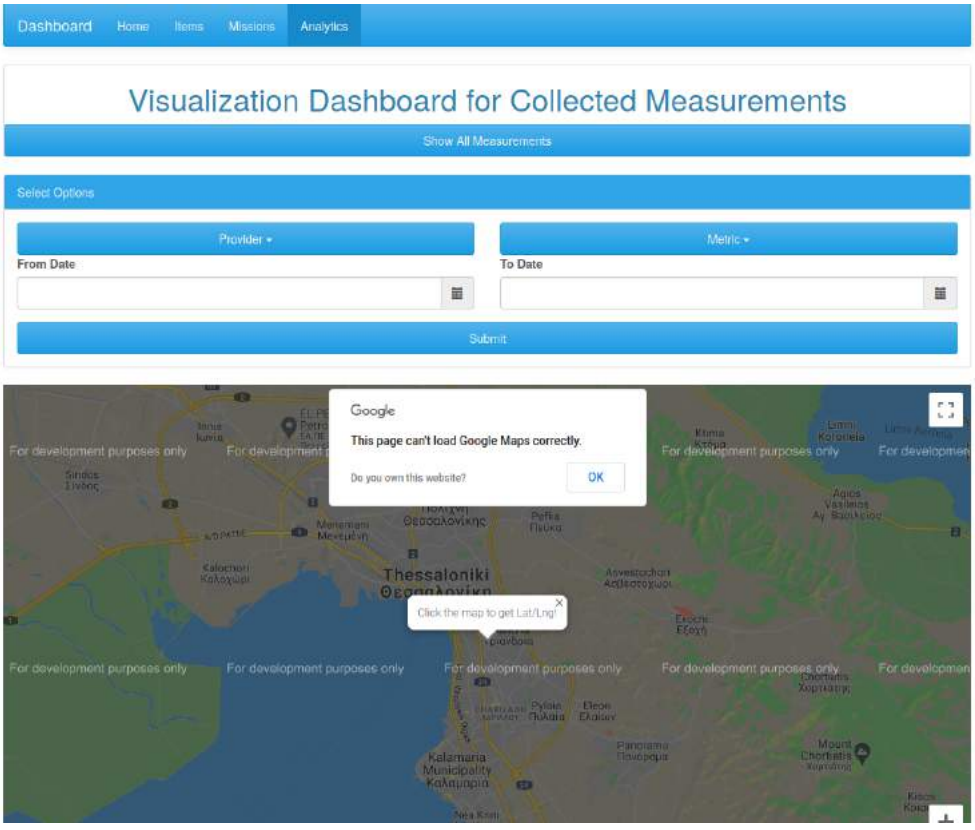


Figure 5.28: Analytics Dashboard

Select a metric that in our case is RSSI.

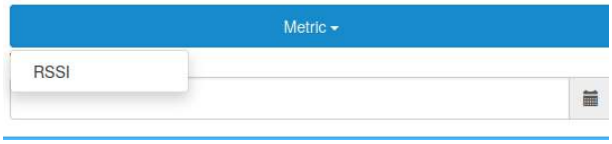


Figure 5.29: Select metric

Select provider from a dynamic list that is populated via our Restful API.

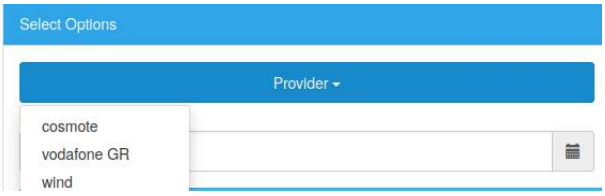


Figure 5.30: Select provider

We show on the left side of the map a table with rssi scaling colors on dBm to help us better understand the selected plotted metrics.

Figure 5.31: RSSI Scale

Plot rssi metrics on a map using coordinates we have collected while player is navigating to collect the items-treasures of the game. All the metrics are plotted in the form of small circles that corresponds to a specific color based on the collected rssi metrics from user's end

device. Also a drawing tool is available from google maps if we want to select a specific area to plot our metrics.

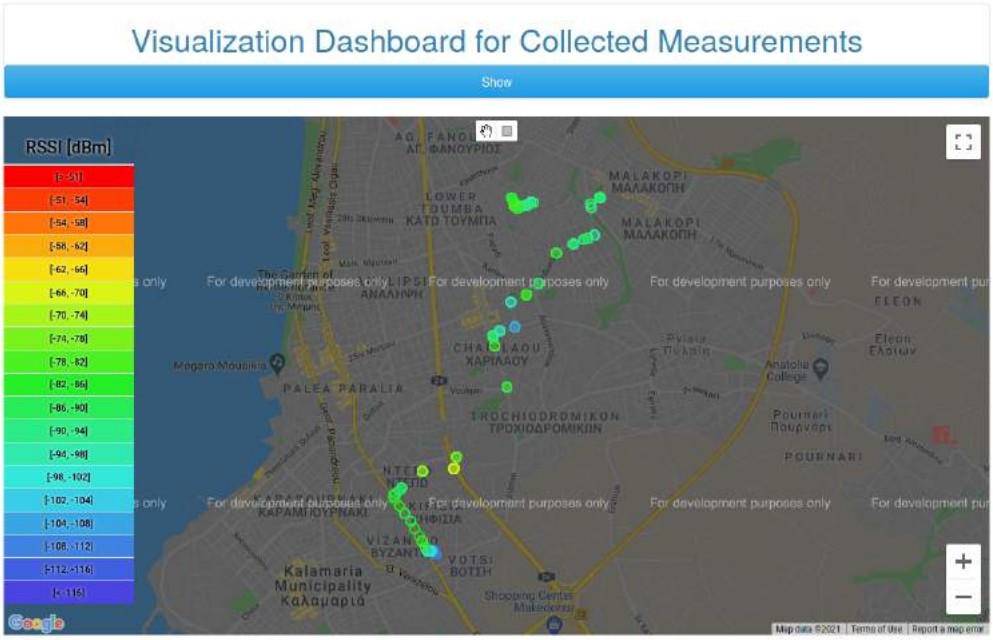


Figure 5.32: Plot RSSI

5.4 Application Architecture Diagram

A technical architecture diagram provides a bird's eye view of the infrastructure of your organisation. The diagram illustrates how components in a system interact with one another in the large scale of thing [55]. As the name suggests, the application architecture diagram focuses on the applications within a computer system and how they interact with databases and each other.

.

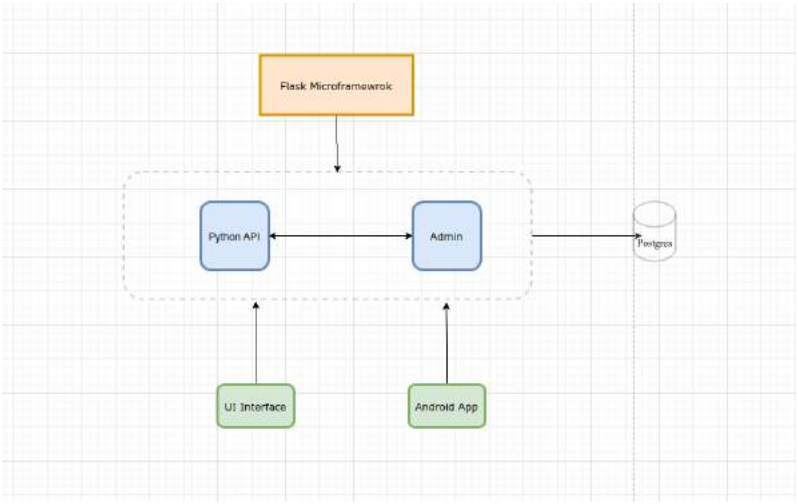


Figure 5.33: Architecture Diagram

An "architecture" can be defined as an abstract description of entities in a system and the re-

lationships between them. It involves a series of decision-making processes. The architecture is a structure and a vision. A ”system architecture” is the embodiment of concepts and the distribution of the correspondences between the functions of things or information and formal elements. It defines the relationships among elements as well as between elements and the surrounding environment. An architectural diagram is a diagram of a system that is used to abstract the overall outline of the software system and the relationships, constraints, and boundaries between components [56]. It is an important tool as it provides an overall view of the physical deployment of the software system and its evolution roadmap.

Chapter 6

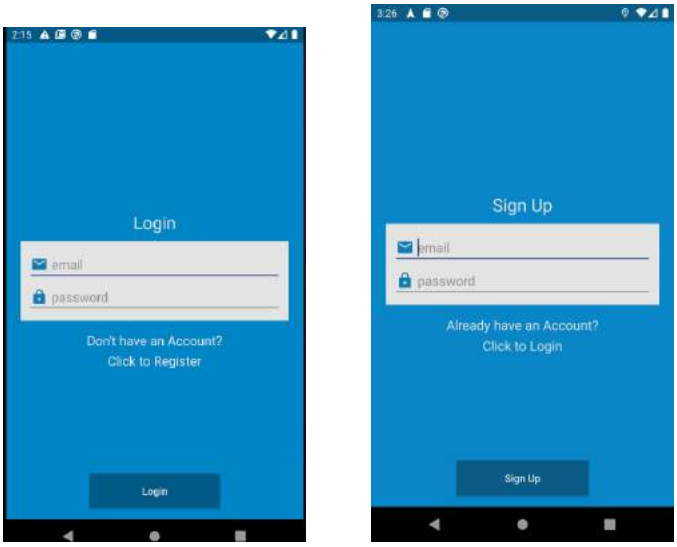
Use Cases - Scenarios functionality

In this chapter we will present different usage scenarios of our application and dashboard area. In more detail we will describe the basic functionality offered by the application through a series of scenarios and images. In these scenarios actions that can performed from the user are presented. This description is indicative of functions performed by the system and does not need to be exhaustive but is primarily purpose is to show how the operations are performed.

6.1 Mobile App Scenario

6.1.1 Player Scenario

John is a new player who just downloaded the application. First he has to register by filling up required data and after that he can logged in to the app 6.1. After registration process he is redirected to the login-screen 6.1 and by filling in his credential he is authenticated and has access to the main screen of the app. Initially a component appears with a short introductory story about the game 6.2.



((a)) Login ((b)) Sign Up

Figure 6.1: Login and Signup images

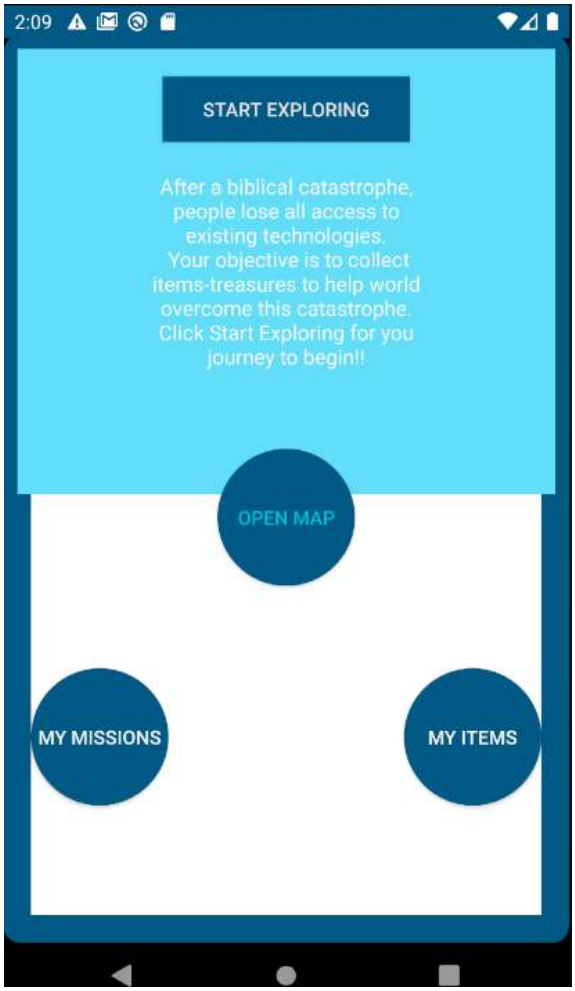
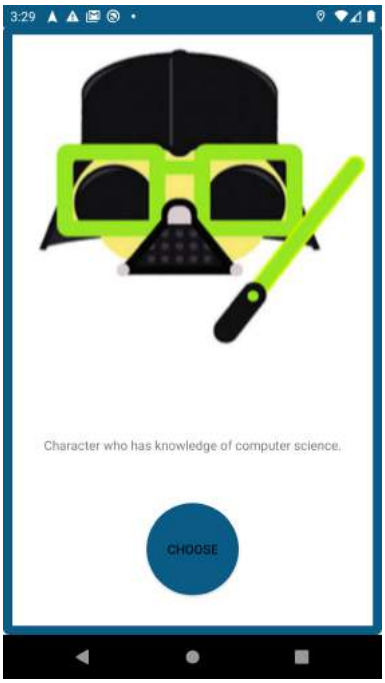


Figure 6.2: description

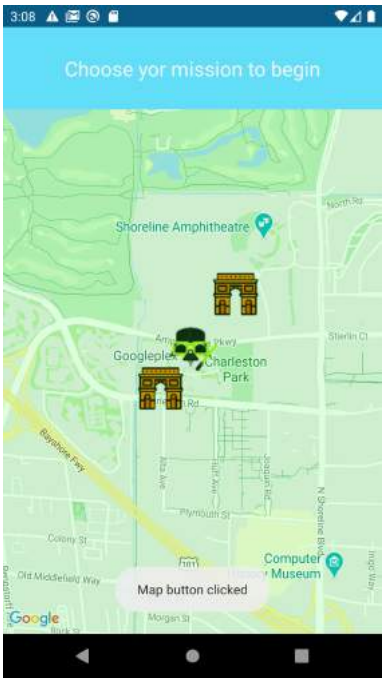
The player has the ability to choose his character after navigating the corresponding page and reading the descriptions to choose whoever he wishes 6.3. Returning to the home page, he can start a mission by clicking the open map button which redirects him to a screen with a map that shows him two possible missions in the form of markers which he can click and read a relevant description 6.4.

((a)) Characters

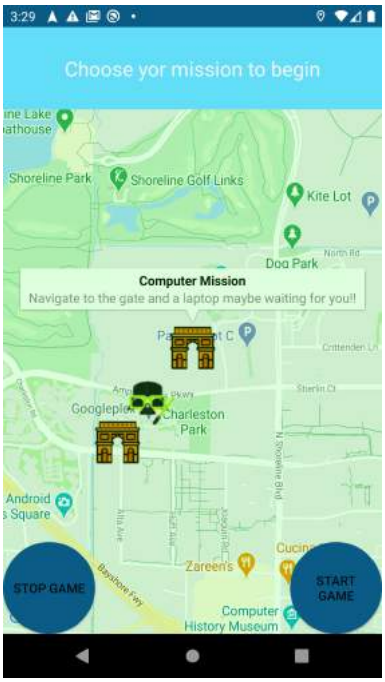


((b)) Character Info

Figure 6.3: Characters views



((a)) Show missions



((b)) Mission description

Figure 6.4: Map views

After choosing his mission and press the start button , google maps navigation system launches 6.5 and he can follows the instructions provided to navigate to the geo-point of interest that may have a treasure item to collect. When he is close to the coordinate (within a radius of 10 meters) a

sound notifies him that he arrived and the app redirects him on the map view and a message with a button collect appears 6.6. User can now press this button and if an item exists can collect it. First mission is just completed!!

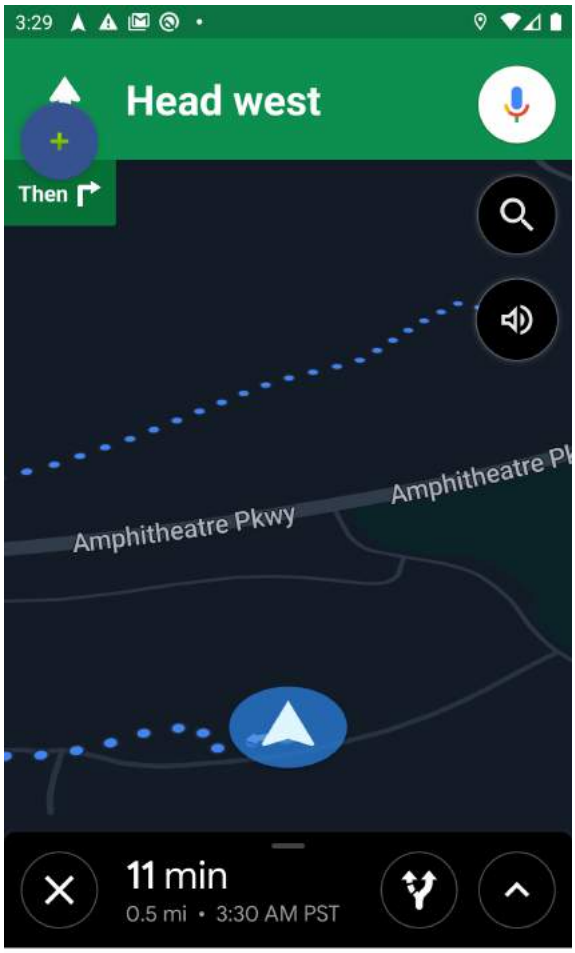
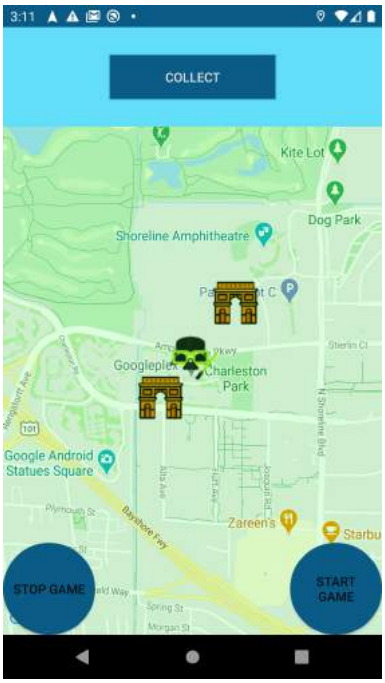


Figure 6.5: Navigation view



((a)) Collect button

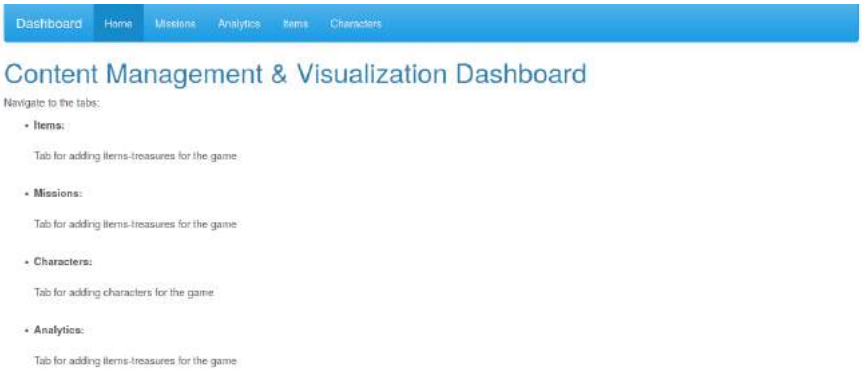
((b)) Collect item

Figure 6.6: Collect item views

6.2 Admin Dashboard Scenarios

6.2.1 Content Management Scenario

John in this case is a system administrator responsible for managing the content of the game. After navigating to the corresponding /admin page 6.7 he wants to add a new mission and a new item. In the missions tab 6.7 he can fill in all the required fields on the form that show up 6.8.



((a)) Admin index

((b)) Missions tab

Figure 6.7: Admin dashboard

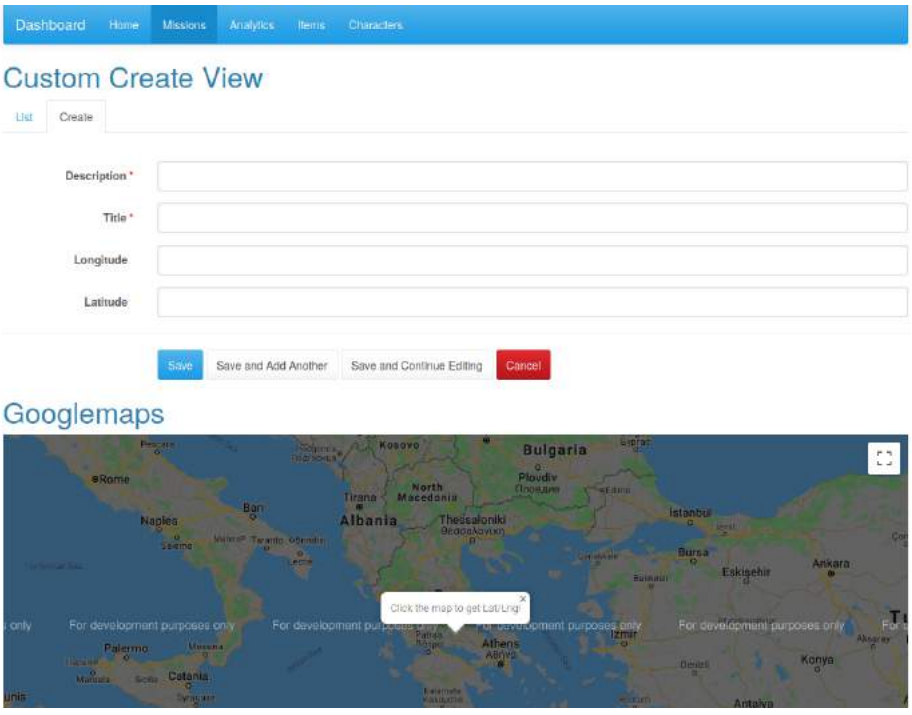
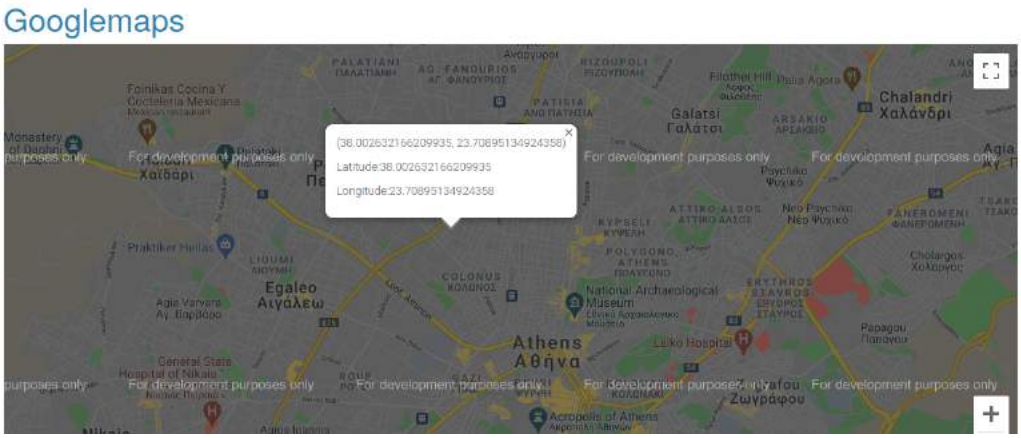
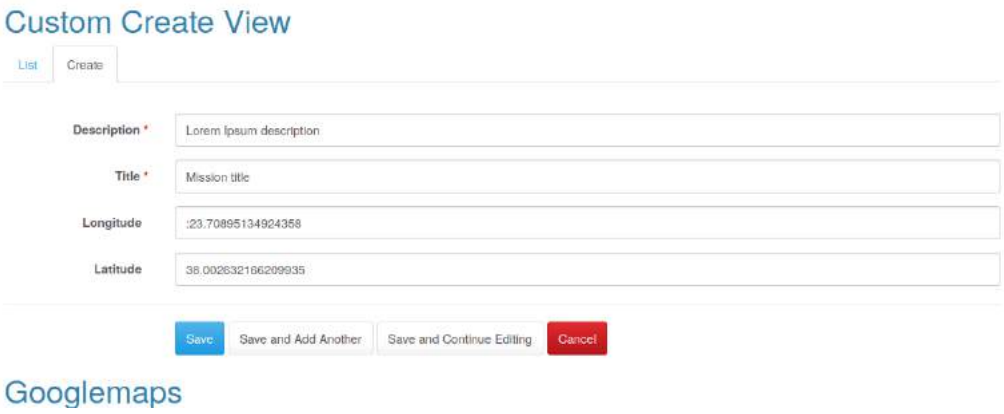


Figure 6.8: Add mission form

In the form he also has the option if he wants to select a point on the map and show him his coordinates. If he wishes, he can fill in the corresponding latitude and longitude fields so that the specific mission appears only to players who will be close to this point. Those coordinates can be retrieved if he click 6.9 a point of interest on the map.



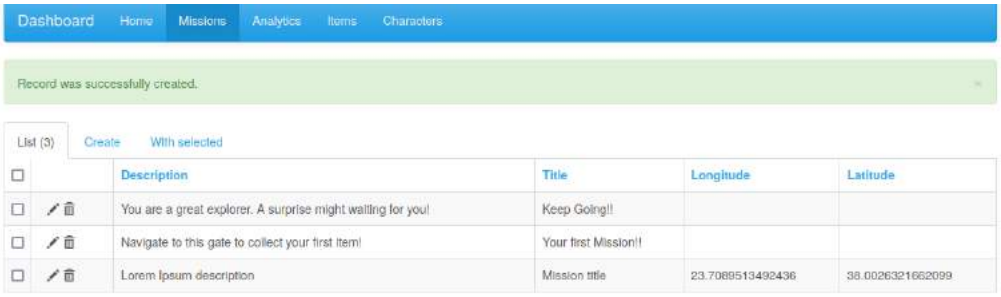
((a)) Click Map



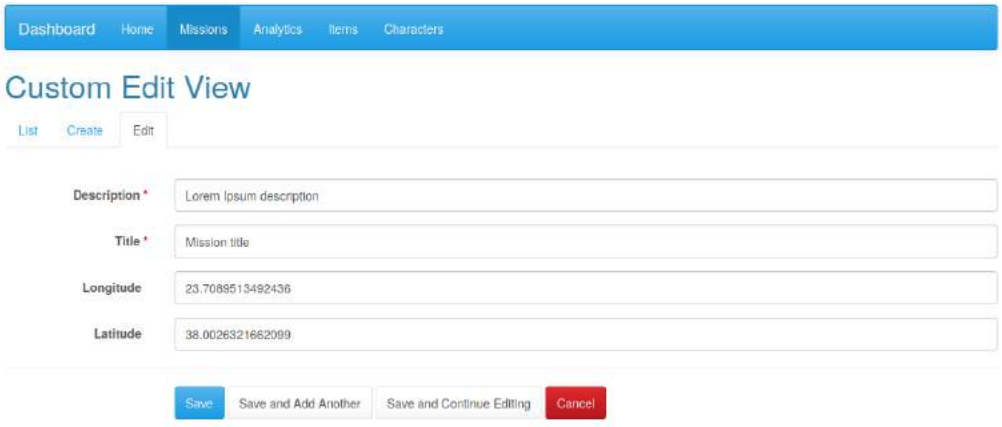
((b)) Mission from

Figure 6.9: Mission Form

By clicking save the specific mission is saved 6.10 in our database and then it becomes available in the game (mobile app). He also wants to add an item. He can also edit 6.10 the mission by pressing edit icon and navigating to edit form if he wants to update mission info. Following the same procedure but in the tab of items respectively can add, view and edit 6.11 an item for the game with title, description and an image.

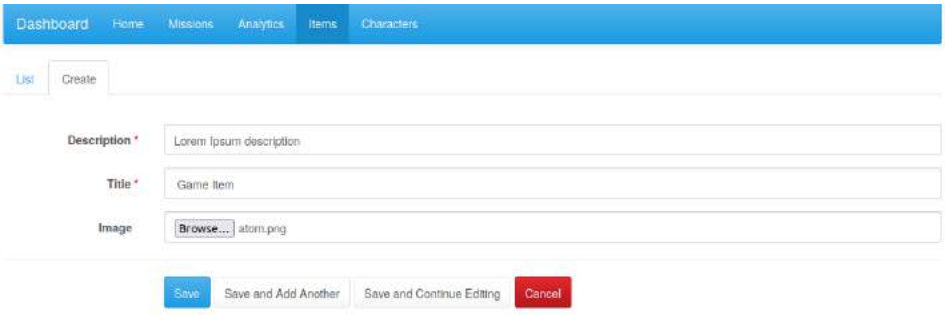


((a)) Success add mission

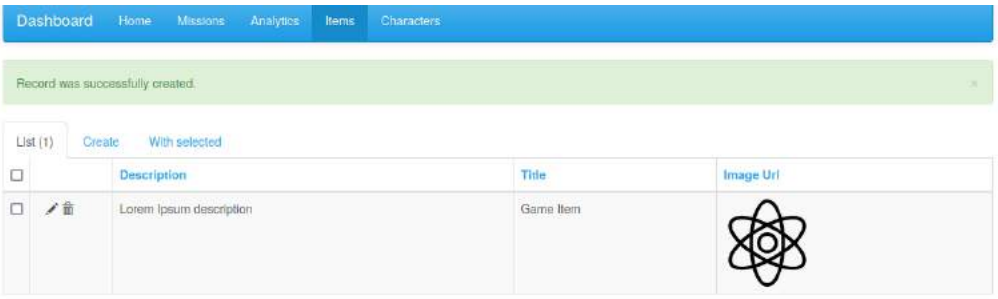


((b)) Edit mission

Figure 6.10: Mission Form



((a)) Create Item



((b)) Item List

Figure 6.11: Item forms

6.2.2 GIS Scenario

George who is an admin user, responsible for the analysis of the measurements and who wants to see the measurements that have been stored in the database for the last 24 hours for a specific provider in a specific area. Going to the analytics tab 6.12 he can select the dates, provider and network type 6.13 for which he is interested in seeing his metrics.

After clicking submit he can sees all the measurements that meet the above conditions plotted 6.14 on the map. He then wants to use the drawing tool, which allows him to select a specific geographical area, which plots a grid 6.15 of specific dimensions and calculates the average of the values in the corresponding squares of the grid.

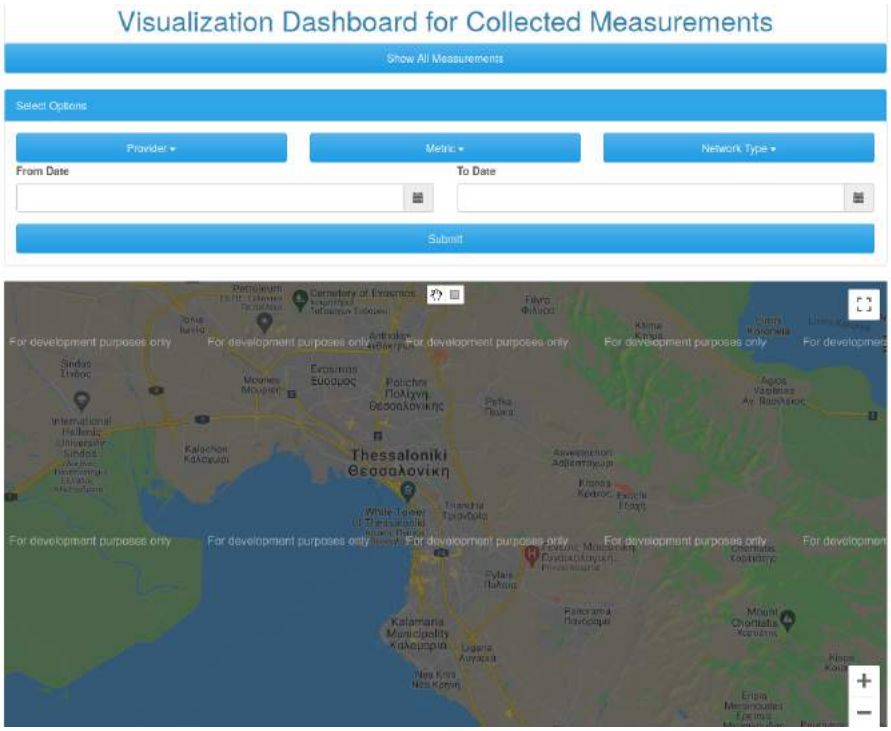


Figure 6.12: Analytics tab

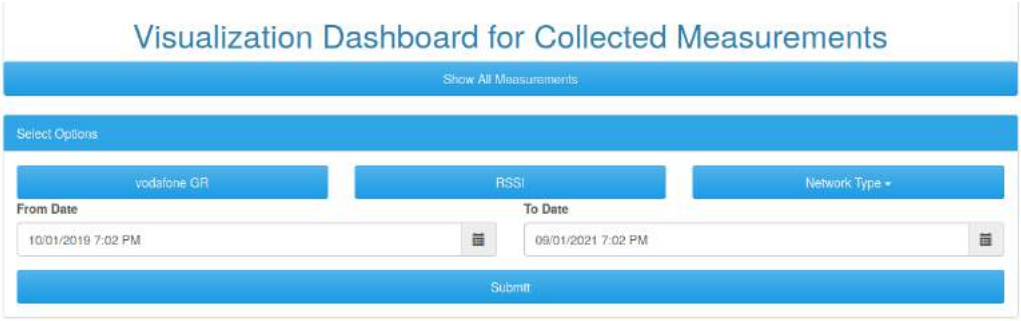


Figure 6.13: Filters

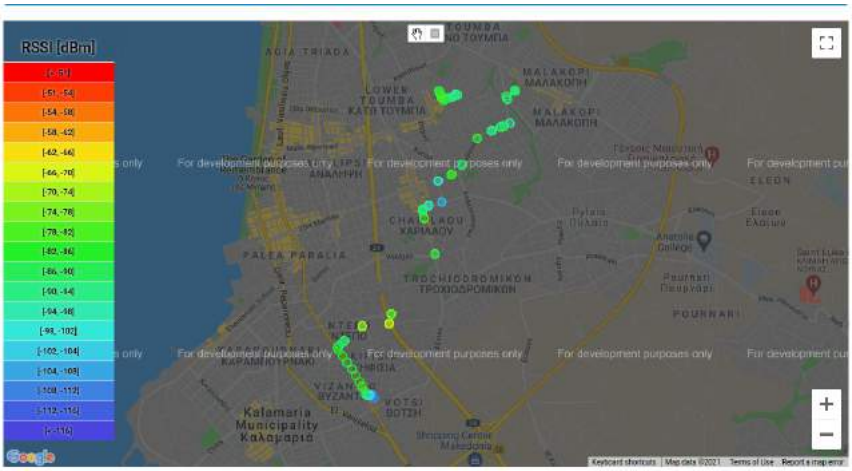


Figure 6.14: Plotted Metrics

Figure 6.15: Drawing tool

Chapter 7

Conclusions

Network providers are in a continuous process to achieve the best possible quality of service for their customers, especially at this time when we are facing the transition to a new generation of networks with the advent of 5G. All telecommunications service providers promise excellent quality of their services and almost absolute population coverage, without necessarily referring to actual geographical coverage.

Companies that provide telecommunications networks rely mainly on external partners to evaluate the quality of services provided and the quality of their networks. Additional proposals for this process have been made either by using smart sensors that could be placed in various parts of the urban environment or through applications for smart devices and the use of crowd sourcing. The main drawback in both cases is that they assume that their end users have knowledge about the metrics they collect.

The system we presented in this thesis - consisting of a game for smart devices and an interactive website - allows us to collect these measurements needed to map the geographical coverage of mobile networks without the users having any knowledge about the collected metrics. The users - players of the game simply play the game completing the missions and the application in the background without requiring any additional action from the user stores the collected data in a centralized system.

We also enable the administrators of the system through the CMS to edit the content of the game and with the appropriate geographical points - missions - to direct players to possible points of interest. This way telecommunication network providers can have a better real-time understanding of the quality of services provided without the need to involve third party companies in this process.

7.1 Improvements and future work

Entering the era of the new generation of 5G networks the most important improvement that could be made is to adapt the mobile application and be able to collect metrics (KPI's) related to these new networks. Below we list improvements that could be made and some ideas for future expansion of the project.

- Add extra KPI metrics for LTE and new generation networks (5G).
- Add functionality for route planning for the game on admin site.
- Admin dashboard for the content management of the game can be used by other applications that are related to geographical planning needs.

- Add interactive-fancy videos in the game flow.
- Improve UI/UX experience of mobile and admin components.
- Complete extra views on android application.
- Add admin authentication based on roles.
- Add Augmented reality functionality for capturing treasures.

Appendix A

LATEX

A.1 Useful links for Latex :

1. Installing Latex on Windows :
   http://www.howtotex.com/howto/installing-latex-on-windows/
2. Installing Latex-Texmaker on Ubuntu :
   - http://tex.stackexchange.com/questions/289458/download-and-installation-of-latex
   - https://www.devmanuals.net/install/ubuntu/ubuntu-16-04-LTS-Xenial-Xerus/how-to-install-texmaker.html
3. Download TexMaker Editor :
   http://www.xm1math.net/texmaker/download.html
4. Tutorial about Latex :
   http://www.tug.org/tutorials/tugindia/chap06-scr.pdf
5. Bold, italics Underlined text :
   https://www.sharelatex.com/learn/Bold,_italics_and_underlining
6. Latex Colors : http://latexcolor.com/
7. How to use colors in latex : https://www.sharelatex.com/learn/Using_colours_in_LaTeX
8. Create your own colors :

   ```
   \definecolor{airforceblue}{rgb}{0.36, 0.54, 0.66}
   ```

9. How to use Hyperlinks :
   https://www.sharelatex.com/learn/Hyperlinks
10. Place image below text with float : http://tex.stackexchange.com/questions/8625/force-figure-placement-in-text
11. Create Tables Online for LATEX :
    http://www.tablesgenerator.com/

12. Source code for listing code with frame : http://tex.stackexchange.com/questions/14967/source-code-listing-with-frame-around-code

13. Listing Caption Numbers : http://tex.stackexchange.com/questions/46464/listings-caption-number

14. Numbers within listing - (After BeginDocument) : http://www.latex-community.org/forum/viewtopic.php?f=5&p=10165

15. Writing code in Latex Document) : http://stackoverflow.com/questions/3175105/writing-code-in-latex-document

Bibliography

[1] Networks history https://www.slideshare.net/gauravsitu/cellular-network-history

[2] Cellular Technologies for Emerging Markets: 2G, 3G and Beyond 1st Edition, Kindle Edition by Ajay R. Mishra

[3] 2G https://en.wikipedia.org/wiki/2G

[4] 3G https://en.wikipedia.org/wiki/3G

[5] 4G https://en.wikipedia.org/wiki/4G

[6] LTE Advanced A Complete Guide - 2020 Edition Paperback – 21 April 2021 by Gerardus Blokdyk

[7] 5G https://en.wikipedia.org/wiki/5G

[8] https://www.monroe-project.eu/monroe-lte/

[9] https://awtg.co.uk/mobile-network-benchmarking

[10] https://www.rohde-schwarz.com/dk/solutions/test-and-measurement/mobile-network-testing/quality-benchmarking/network-benchmarking_231992.html

[11] https://www.nfon.com/en/service/knowledge-base/knowledge-base-detail/qos

[12] Quality of Service Regulation Manual", ITU, 2017, available at: https://www.itu.int/dms_pub/itu-d/opb/pref/D-PREF-BB.QOS_REG01-2017-PDF-E.pdf

[13] https://en.wikipedia.org/wiki/Quality_of_experience

[14] Panagiotis T. Koziokas, Nikolaos D. Tselikas, George S. Tselikis, "Usability Testing of Mobile Applications: Web vs. Hybrid Apps", in proceedings of 21st Panhellenic Conference on Informatics, PCI 2017, Sept. 27-30, Larisa, Greece https://dl.acm.org/doi/10.1145/3139367.3139410

[15] https://iopscience.iop.org/article/10.1088/1742-6596/1734/1/012031/pdf

[16] Crowdsourcing https://crowdsourcingweek.com/what-is-crowdsourcing/

[17] https://www.investopedia.com/terms/c/crowdsourcing.asp

[18] https://www.livemint.com/Companies/L1AD703BID2j16c4HMzYcK/How-businesses-should-use-crowdvoting.html

[19] https://cse.buffalo.edu/~jing/doc/CIKM16.pdf

[20] https://en.wikipedia.org/wiki/Crowdsourcing

[21] https://www.clickworker.com/crowdsourcing-glossary/mobile-crowdsourcing/

[22] https://www.researchgate.net/publication/294424491_Mobile_crowdsourcing_Framework_challenges_and_solutions

[23] https://www.researchgate.net/publication/329492788_A_Review_of_Mobile_Crowdsourcing_Architectures_and_Challenges_Toward_Crowd-Empowered_Internet-of-Things

[24] https://en.wikipedia.org/wiki/Android_(operating_system)

[25] https://www.androidauthority.com/what-is-android-328076/

[26] https://www.tutorialspoint.com/android/android_architecture.htm

[27] https://www.tutorialspoint.com/android/android_acitivities.htm

[28] https://www.journaldev.com/13639/retrofit-android-example-tutorial

[29] https://www.python.org/doc/essays/blurb/

[30] https://pythonbasics.org/what-is-flask-python/

[31] Flask Creator: https://github.com/mitsuhiko

[32] Jinja: https://jinja.palletsprojects.com/en/3.0.x/

[33] WSGI: https://www.python.org/dev/peps/pep-0333/

[34] https://werkzeug.palletsprojects.com/en/2.0.x/

[35] https://flask-admin.readthedocs.io/en/latest/

[36] https://whatis.techtarget.com/definition/Google-Maps

[37] POSTGRES Project : https://www.postgresql.org/about/

[38] POSTGIS : https://postgis.net/

[39] https://www.postgresql.org/about/

[40] https://www.tutorialspoint.com/html5/html5_overview.htm

[41] https://www.w3schools.com/js/

[42] https://www.w3schools.com/css/

[43] https://swagger.io/

[44] https://github.com/flasgger/flasgger

[45] https://www.sitepoint.com/use-ngrok-test-local-site/

[46] https://www.blazemeter.com/blog/how-use-postman-manage-and-execute-your-apis

[47] Geoffrey Elliott Josh Strachan (2004) Global Business Information Technology. p.87.

[48] SDLC: https://www.softwaretestinghelp.com/software-development-life-cycle-sdlc/

[49] SDLC (Replace): https://www.softwaretestinghelp.com/software-development-life-cycle-sdlc/

[50] http://www.unity3dtechguru.com/2018/01/gdlc-game-development-life-cycle.html

[51] http://www.unity3dtechguru.com/2018/01/gdlc-game-development-life-cycle.html

[52] https://www.nuclino.com/articles/game-design-document-template

[53] https://www.klipfolio.com/resources/articles/what-is-data-dashboard

[54] https://en.wikipedia.org/wiki/Create,_read,_update_and_delete

[55] https://www.leanix.net/en/technical-architecture

[56] https://www.alibabacloud.com/blog/how-to-create-an-effective-technical-architectural-596100