Communicate to Win

Real-time communication services for location-based learning activities

Author: Amir Buchvalter
Supervisor: Prof. Miky Ronen
Co-supervisor: Dr. Dan Kohen-Vacs
Examiner: Dr. Nuno Otero
Date: 30th May 2017
Subject: Social Media and Web Technologies
Level: Master
Course code: 5ME11E
Abstract
In recent years, mobile devices have become an integral part of our everyday life in various fields. The technology that powers them is used in various devices, such as smartphones, tablets, and PDAs. These devices hold extensive computation capabilities, along with advanced communication abilities, which are supported by an internet connection and diverse types of motion and location sensors. Mobile devices have changed the way people communicate with each other, for example by providing options to send instant text messages, perform live video calls with others in addition to voice conversations.

These capabilities have encouraged educators to exploit mobile technology and promote new types of learning options. There are several new learning possibilities based on tools such as mobile dedicated applications, location-based learning activities, and interactive social related tasks. These new uses require adjustments to educational programs to allow support for this type of learning. The uniqueness of mobile learning, in contrast with the classic learning paradigm, is the ability to connect the learner through enhanced learning materials to the outside environment. This breaks the physical borders that exist in the traditional classroom and creates new learning possibilities, but it has its drawbacks. One of the difficulties that arise from this type of learning is the loss of contact in real time between teacher and student, especially when performing outdoor activities.

The Treasure-HIT platform was conceived to create outdoor Treasure Hunt-based games, mainly for educational purposes, by introducing an authoring web platform and a supporting mobile application. The platform includes communication capabilities based on social interactions and cooperative learning with the integration of social networks, and yet real-time educator and student communication is still lacking. Adding real-time communication features provides a way to better support ongoing learning activities, and can take pressure off the learning process for the students, by providing them with a more personal experience and immediate support when needed.

This thesis follows the process of addressing this concern in Treasure-HIT, including the rationale, the background, the possible options and gradual development of a prototype solution to the problem under the existing Treasure-HIT infrastructure.

In the thesis, the advantages of two features are explored: (1) the Real-time Group Tracking Map, which provides a way to monitor the movement and action of groups of players during game time, and (2) Instant Text Messaging service, which allows the game instructor to send custom text messages to the different game groups.

The research findings indicate that the new monitoring options provide a solid tool for real-time analysis of the progress of the game and the ability to inform about various issues and solve them in real time.
Furthermore, the instant message service feature received positive responses from the game-manager and players alike, on the grounds of major improvements to the general game flow and problem solving in real-time.

**Keywords**
Real-time communication, mobile learning, location-based learning, outdoor learning treasure-hunt

**Acknowledgments**
I would like to express my deepest gratitude to my supervisor, Prof. Miky Ronen, for the continuous support, understanding, personal sensitivity and motivation during my academic studies and especially in this thesis.

I would also like to express my sincere gratitude to my Co-supervisor, Dr. Dan Kohen-Vacs, for his assistance and ongoing support throughout my entire graduate studies. I appreciate the patience, personal attitude and all the insightful technical and pedagogical comments which helped to shape the thesis and development efforts in the best way.

Many thanks to my friends In the Instructional Design Department at Holon Institute of Technology, especially to Rotem Israel and Reut Bachar, for the tremendous support in pedagogical matters, technical implementation and motivation. This thesis could not be done without you.

To my beloved wife, Shani Band, who gave me the space and time during the long hours of development and research, who is always willing to listen and give good advice and to provide help of any kind when needed.

Finally, I would like to thank my lovely family, my parents and brothers for their love and support throughout the writing process. I would especially like to thank my father, Immanuel Buchvalter, who once told me that a person who has no curiosity will achieve nothing.
# Table of Contents

1 Introduction and Motivation ........................................................................................................1

2 Literature Review .......................................................................................................................2
   2.1 Technology-Enhanced Learning .........................................................................................2
   2.2 Location-based Learning ..................................................................................................5
   2.3 Location-based Games .......................................................................................................7
   2.4 Real-time communication for computer based learning ...................................................8

3 Introduction to The Treasure-HIT Platform .................................................................................10
   3.1 Activities Editor ................................................................................................................11
   3.2 Mobile Application ..........................................................................................................13
   3.3 Treasure-HIT Architecture and Technology .......................................................................15
      3.3.1 Service Orientated Architecture ..............................................................................15
      3.3.2 MVC Design ............................................................................................................16
      3.3.3 Treasure-HIT web application architecture ..............................................................17
      3.3.4 Treasure-HIT mobile application architecture ..........................................................19

4 Research Questions .....................................................................................................................21

5 Methodological Approach .........................................................................................................21

6 Development and Evaluation Plan ..............................................................................................22

7 Implementation ...........................................................................................................................23
   7.1.1 Basic Perceptions ............................................................................................................23
   7.1.2 Interviews with Key Authoring Expert Users .................................................................24
      7.1.2.1 Field Observation ..................................................................................................25
   7.2 Phases 2 and 3 - Design, Development and Testing ..........................................................26
      7.2.1 Real-time groups’ tracking map ..................................................................................26
         7.2.1.1 Design ..................................................................................................................26
         7.2.1.2 Development (technical aspects) ...........................................................................28
         7.2.1.3 Testing and Refinements .....................................................................................34
         7.2.1.4 Field Observation .................................................................................................38
      7.2.2 Game-manager and Player Instant Messaging service .................................................39
         7.2.2.1 Design ..................................................................................................................39
         7.2.2.2 Development (technical aspects) .........................................................................41
         7.2.2.3 Testing and Refinements .....................................................................................43
         7.2.2.4 Field Observation .................................................................................................44

8 Discussion .......................................................................................................................................47

9 Summary and Conclusions .........................................................................................................49
Table of Figures

Figure 1. Games activity grid ................................................................. 12
Figure 2. Game properties settings ....................................................... 12
Figure 3. Station properties editor ........................................................ 13
Figure 4. Access a game by ................................................................. 14
Figure 5. Location GPS verification ....................................................... 14
Figure 6. Multichoice question presentation .......................................... 14
Figure 7. Feedback message .................................................................. 14
Figure 8. Clues message ...................................................................... 14
Figure 9. End game message ............................................................... 14
Figure 10. Game screens workflow ....................................................... 14
Figure 11. MVC tiers of the Treasure-HIT web and mobile applications ...... 17
Figure 12. Sequence diagram of the Treasure-HIT authoring system, new station dataflow ................................................................. 18
Figure 13. Sequence diagram of the Treasure-HIT Mobile application, Mobile device and backend components interaction .................................................. 20
Figure 14. Design based research workflow .......................................... 22
Figure 15. Treasure-HIT activity on H.I.T orientation day ....................... 24
Figure 16. Groups' activities map navigation button ............................... 27
Figure 17. Stations data types in-map presentation ................................. 27
Figure 18. Real-time group tracking map UI mockup ................................ 28
Figure 19. New game groups location map navigation button under games grid ................................................................. 28
Figure 20. Routing action to the Dynamic Map controller ........................ 29
Figure 21. GameGroupData ViewModel ................................................ 29
Figure 22. GetGroupsNameAndIds controller method ............................. 30
Figure 23. CreateActivityPointsJson controller method .......................... 30
Figure 24. Map center termination by point array values .......................... 31
Figure 25. Activities stations indication and information window .............. 31
Figure 26. Groups locations fetching controller method .......................... 31
Figure 27. Mobile application API call for group's activity logs registration ................................................................. 32
Figure 28. Sequence diagram of the Real-time groups' tracking map feature backend data flow .................................................................................. 33
Figure 29. PolyLines data entry for group estimated path creation ............. 35
Figure 30. Group Activity points estimated path display ........................... 35
Figure 31. Map and Reduce inner loop for station and group activity points ................................................................. 36
Figure 32. Draw polylines on the map between stations and relevant activity points ................................................................. 36
Figure 33. Group activity rally lines drawn to related station ................. 37
Figure 34. New calls to generalHandler factory class methods ............... 37
Figure 35. Group activities information window ..................................... 38
Figure 36. Registration of wrong location check points due to a GPS reception issue .. 39
Figure 37. Registration of wrong location check points due to an orientation problem ................................................................. 39
Figure 38. Sequence diagram of the custom messages service dataflow ...... 41
Figure 39. New message data encapsulated JavaScript function .......... 41
Figure 40. API call for the selected group message text data .................. 42
Figure 41. API call to get text message from backend ........................... 42
Figure 42. Mobile application user activity action types .......................... 43
Figure 43. Game-manager Message alert popups after a "check answer" user action... 43
Figure 44. Check location with get message API call ................................ 44
Figure 45. Player Indications of the progress of the group and the provision of an estimated route on the map................................................................. 45
Figure 46. Game-manager, while monitoring a group progress in real time.......... 45
Figure 47. Game-manager message displayed over a player's smartphone .......... 45
Figure 48. Read indication is set on false..................................................... 46
Figure 49. Read indication is set on true ......................................................... 47
Figure 50. Character count textbox addition ............................................... 47
Figure 51. Message textbox control with character count function call.............. 47
Figure 52. Supervisor Evaluation Questionnaire......................................... XV
Figure 53. Player Evaluation Questionnaire................................................ XVI

List of Tables

Table 1. Evaluations summary ........................................................................ 51
1 Introduction and Motivation

In recent years, modern technology has led to significant changes in the field of learning environments, both from the instructor’s and learner’s perspectives. Mobile devices have become more and more convenient and affordable with every new release (Zydney & Warnet, 2015), which has caused them to gain popularity and became part of everyday life (Wong, 2014). As a result, new learning methods have been developed in order take advantage of the quality and the mobility of these devices. These methods enable users to learn anytime and anywhere, without the normal constraints of the traditional classroom (Zydney et al., 2015). One of the major emphases is context awareness for learning purposes, meaning the learning activity is not just a plain task with questions to answer but rather a dynamic learning process that takes advantage of the outdoor learning environment.

The mobile nature of the modern devices allows educators to exploit it for the creation of new tools and methods, while taking advantage of the GPS component and other sensors embedded within the different modern mobile devices (Avouris & Yiannoutsou, 2012). Some of the innovative approaches which take advantage of those features, are location-based learning activities and games. Mobile games exist in many forms and vary from treasure-hunts to location-based games, scientific applications with physical context awareness, and language learning helpers (Sun & Chang, 2016). Most of those apps encapsulate principles of synchronous communication, which are realized under different real-time learning experiences. These communication abilities vary form instant messaging options, location recognition and sharing, live chat, VOIP protocols, and live video streaming.

The Treasure-HIT platform, whose development started in 2012 and is ongoing, is an example of a system that allows the playing of location-based learning games while using mobile devices. The platform consists of a game editor and a mobile application for treasure hunt games, that are especially designed for educational purposes (Cohen, 2015). Thanks to the smartphones’ and tablets’ location abilities, the Treasure-HIT mobile application can determine the user’s location and allow the game creator to create location-based activities that take place outside of the classroom. These types of learning activities, such as those based on the current Treasure-HIT platform, where there is physical separation between the game-manager and the student, may lead to lack of guidance and help when most needed. Thus, communicating feedback and keeping track of the proceedings on the game field is crucial. Exploiting the devices’ built-in GPS component and the ability to keep an open internet connection during game time allows the game-manager to communicate with the students to post and get relevant activity information. The importance of the platform’s communication capability has been emphasized in previous researches (Israel, 2016), which indicated the need for real-time communication services in order to provide a better experience to the learners, as well as to increase and refine the game-managers’ choices while monitoring the game and communicating with the participants. Therefore, it was decided in this study to develop advanced communication capabilities, as detailed in the following chapters.
# 2 Literature Review

## 2.1 Technology-Enhanced Learning

In recent years, technology has become a major component of people’s everyday life, in a large spectrum of activities, including learning (Trepule, Tereseviciene & Rutkiene, 2014). The fast growth of information and communication technologies (ICT) has created various platforms of learning management and online learning environments (Chen & Lin, 2011).

Technology-Enhanced Learning (TEL) is a term that usually describes a system that strives to find a technological solution for educational problems, bringing together technology and pedagogy. The system is based on engagement with different communities, and the use of technology depends on the usages and practices of designing and developing educational solutions (Scanlon, O'Shea & McAndrew, 2015).

Trepule and others (2014) have provided a description of several characteristics of the TEL paradigm that relate to the learner’s role:

- The learner should be at the center of the educational activity, from the initial step of the educational activity design, to the practice of the technological learning system.
- The learner should have different possibilities to engage specific topics.
- The learner should have options for collaboration with other students and with the instructor.

TEL presents an opportunity to enhance the learning process of the learners, while providing the ability to expose the learner to a vast amount of information, the majority of which would not have been available if not for the internet and its ability to communicate on different levels, such as communication of learner-to-learner, learner-to-instructor and instructor-to-instructor. These, in turn, enable direct and fast feedback, sharing of ideas and solutions, collaborations and general support during the learning process (Blanco, Van der Veer, Benvenuti & Kirschner, 2011). These different social interactions have a major influence on the learner’s motivation and achievements (Trepule et al., 2014).

There are numerous examples of technology enhanced learning systems, among which are online courses, technology-based learning activities and games. One of the most popular learning support architectures, which is based on the properties of TEL, is the Learning Management System (LMS). Those type of systems, act as a tool for managing learning activities, that use features such as tracking a learner’s progress, or maintaining a Database for storing and managing learning contents (Ji, Park, Jo & Lim, 2016). The Treasure-HIT platform, which constitute the infrastructure for this research developments, is an example of a type of an LMS system. The Treasure-HIT system, can create and manage learning content, present it to the leaner in a form of a mobile game and get feedback its participants, but as mentioned in the introduction sections, in its current state, the system is lacking communication tools which is one of the key elements of LMS systems.
One example of an LMS system is Moodle, which is an online open source system that supports blended-learning methodologies with synchronous and asynchronous communication abilities. It is widely adopted by various educational environments, organizations (Macho & Robles, 2013) and universities worldwide (Fernández-Robles et al., 2013). Moodle is mostly used to manage online courses, accommodated knowledge, Wikis, online learning materials, discussion forums and direct communication between the system users. Also, by nature of being an open source platform, Moodle enables adding customizable functions to its basic state in the form of software plug-ins (Zacarias et al., 2016), and thus to add additional possibilities that would best suit the type of learning, content, and specific learner.

Another possible implementation of a TEL is an online learning activity. An online activity was conducted by Chen and Lin (2011), whose purpose was to research the motivational factors of learners during a course of product design. The course goal was to enable the students to acquire design principles and techniques. The activity was based on an online design learning environment that was self-developed by the researchers. It contained online education material usage, course information sharing by the instructors, communication via live chat and video, and sharing of ideas by submission of media files to the course’s forums.

Computer games constitute one of the most popular implementations of TEL. Ardito, Costabile and Lanzilotti (2010) suggest that technologically engaged games can act as a major motivational tool for learners during educational activities. In the researching best practices of educational games, the researchers created two new computer games designed to teach about historical sites. The games were especially designed for middle-school age pupils. The pupils were required to solve various puzzles and play trivia games on a mobile device. In this specific case, the mobile device came from the special UI game interface that allowed multitouch actions in order to solve the challenges presented by the game. These examples show some of the possibilities for extensive and original user interactions through mobile devices, which will be detailed and explained in the following sections.

Over time, mobile phone usage has expanded into various fields. They are used in public services, banks, cultural facilities, museums, workplaces and educational institutions. The fact that these devices are based on mobility makes them perfectly suited for educational purposes. Mobile phones provide students with the possibility of learning within and beyond the classic classroom (Karimi, 2016).

Mobile learning is defined as a learning process which occurs on mobile devices such as smartphones, PDAs and tablets. The use of mobile devices allows learners to learn anywhere and anytime (Karimi, 2016), regardless of the learner’s location, whether they are concentrated in one place or whether they deployed over a wide area (Miguel, Caballé, Xhafa, Prieto & Barolli, 2016). Mobile learning, in its essence, avoids the restrictions of the physical traditional classroom and provides a more autonomous learning experience (Teodorescu, 2015). Mobile learning, based on smartphones and tablet devices, also allows educators to redesign the way the learners will learn. The educators can create a more communicative and flexible environment that makes multiple data sources available in a hand-held device that every learner has, to create a less authoritarian learning process, one that is established on mutual communication between students and educators (Heflin, Shewmaker & Nguyen, 2017).
One of the implementations of mobile learning on mobile devices comes in the form of mobile applications. Mobile applications used for educational purposes can vary from humane and scientifically oriented apps to various educational games (Domingo & Gargante, 2016). “Moin” is a mobile application that was developed by the researchers Ngan, Lifanova, Jarke and Brober (2016). The application focuses on social and cultural assistance factors for refugees arriving to Germany since the year 2015. The application’s aim is to encourage young refugees, most of whom have a mobile phone with an internet connection, to assist in the immigration process by addressing cultural barriers, communication difficulties and gaping cultural differences.

“Great STEM”, is a mobile application which was designed to encourage children to explorer different domains of science. The application was designed to enable the exploration of the physical space of a science museum and interact with its different exhibited objects. The students, during learning activity, are required to scan different QR codes, which trigger different challenges on the mobile device for the students to handle. These challenges are focused on the fields of technology, engineering and mathematics. Later, the students can share their collected media among fellow students and intact using comments and “Likes” (Atwood-Blaine & Huffman, 2017).

Furthermore, the mobile device’s hardware allows developers and educators to take advantage of the different available technological qualities to enhance the experience for specific personal learning activities. For example, “MobileParsons” is a mobile application designed to educate computer science students with different computer programming questions regarding “parsons problems”, which are based on a set of randomly ordered code fragments and challenge the learners to rearrange them in an order that meets the desired requirements (Helminen, Ihantola, Karavirta & Alaoutinen, 2013). The user’s interactions with the application are based on a graphic user interface (UI), which was specially designed for mobile operations over the device’s touch screen (Karavirta, Helminen & Ihantola, 2012). During the game, the student is presented with an assignment whose solution requires the student to move different code fragments around the screen and place them in the right order. Automatic feedback is provided to the user in the form of text hints. The decision to use a mobile device for this type of exercise derives from the approach by which mobile devices with embedded touch screens are suited for drag and drop actions and for rapid learning sessions (Karavirta et al., 2012).

Not only the usage of mobile devices has significantly increased in recent years, but there is also a growing usage of mobile devices with open access to the internet (Pereira & Rodrigues, 2013), which enables mobile learning to become more interactive due to communication and collaboration with different people (Wong, 2014). The new emerging ways to communicate have transformed learning from an individual experience to a process that requires social practices (Flores & Maciuszek, 2013). Social interactions using mobile devices can enrich the learning experience by creating shared information entities, such as locations, social network enhanced context, and direct communication with other learners and educators (Flores & Maciuszek, 2013).

Mobile learning has many benefits, and provides the learners with an opportunity to experience learning from a different perspective. Some of the benefits are increased engagement levels for tasks accomplishment, autonomous learning in the sense of setting personal learning goals, and collaboration when instruction and help are needed from other classmates (Domingo & Garganté, 2016). Alongside the many advantages of mobile learning, it also has a few disadvantages that were brought up by a few
researchers. Bdiwi and Bargaoui (2015) suggest that although mobile learning can provide access to high quality content using multimedia contexts, some of it might not be always available. Some of the uploaded materials are designed to be presented on a specific operating system, which in turn may cause this content to be unsupported on other devices. For example, Adobe Flash is inaccessible under Apple iOS, whereas Quick Time videos are not supported by Google Android operating systems.

Mobile learning embodies the principle of sharing personal information – such as learning content and physical locations that can be easily accessed by the device’s GPS, personal information, social interactions and communication with various parties both within and outside the learning system (Shuib, Shamshirband & Ismail, 2015). This in turn requires developing solutions to security issues that might arise from this learning paradigm. Miguel and others (2016), define the phases that should constitute an educational activity design, which include sharing of personal information and social interaction. The design emphasizes determining the context of the interaction between parties, setting rules for interactions, enforcing policies for each type of interaction and determining if the engaged entities in the activity are trustworthy, by considering the given interaction context. In order to protect location-based information (collected for example when performing an outdoor learning activity), different educational portals that host educational mobile applications restrict the collecting of such data outside the educational activity itself (Shuib et al., 2015).

2.2 Location-based Learning

Today, modern mobile devices, such as smartphones and tablets, include the ability to receive information about the current location of their holders, using the Global Positioning System (GPS) component, which is based on the availability of a solid satellite reception (Feldman, Sung, Sugaya & Rus, 2015). This feature should be taken into consideration when developing an educational activity for mobile learning on mobile devices (Frank, Lackes & Siepermann, 2015). Mobile devices can improve the perception of the environment in which the learner operates and can connect to informal learning and learning-oriented activities outside of the classic classroom (Harley, Poitras, Jarrell, Duffy & Lajoie, 2016).

Location-based learning is derived from the realm of the context-aware learning. Context-aware learning refers to a situation in which the learner’s physical environment is used as a part of the learning process, and has a direct impact on the learner's capability to deal with the given situations which are related to the learning context of the learner’s physical surroundings (Sun & Chang, 2016). Participants in mobile learning activities are usually required to move around in a physical location that was chosen by the activity’s designer, for example, to identify objects in these locations in the context of the study topics.

Research conducted by Sun and Chang (2016) location-based developed and experimented using a mobile application that educated users about plants, with a focus on scientific and biological terms in the English language. The application directed the user to a specific location in the university campus, that was predetermined by the application’s creators. This exploratory activity succeeded due to the users’ ability to locate the plants by taking advantage of the integrated GPS component on their mobile devices, with the assistance of a cellular network connection, which improved the
accuracy and positioning capability of the device, and the ability to track the learners and assist them on the way. The application presented the learner with the options to fetch information about specific plants and their biological structure, to practice the learned material using an English vocabulary via tools like text presentation over the app’s UI, and to get immediate feedback from the mobile application’s integrated information.

Another interesting development in the location-based learning field is the use of both location and augmented reality concepts which create a relatively new combination that makes for a new learning experience. This type of mobile learning experience, augmented reality on mobile devices, was described by Hürst & Wezel (2011) as a situation where mobile device users look at live images that are streamed using the device’s video camera, and enriched by virtual 3D objects. One example of such a combination was studied by Harley, Poitras, Jarrell, Duffy and Lajoie (2016) and presented in their article *Comparing virtual and location-based augmented reality mobile learning: Emotions and learning outcomes*. The research’s goal was to explore positive emotions in users who used augmented reality applications during a learning context. One of the applications the researchers used in this research was the MTL Urban Museum application, which was developed by the McCord Museum. This application allows users to scan the city of Montreal and explore historic sites from the 1800s. The exploration is done by the creation of interactions with different location contexts using the mobile device’s GPS. Thanks to an integration with Google Earth API, the app can make a comparison of the sites’ appearance then and now, and relates them to the learning materials and relevant contents of the Museum. Another application used in this research was *metaGuide*, which was adopted by the same museum. This application was designed to focus on tour guides and their instructors. Its main function is to pinpoint different locations by GPS indications, and to analyze their appearance by embedded graphical tools that are based on Google Street images and the application’s Database. This ability introduces the museum’s guides personally and allows them to create different context-related reminders and remarks in a specific location’s, create learning objectives, and enable their use in an ongoing guided tour.

LEMONADE is a field trip framework, developed and researched by Giemza, Bollen, Seydel, Overhagen and Hoppe (2010). The framework provides a full cycle authoring platform to manage field trip activities and a mobile application to conduct the field trip activities for the participants during the trip itself. The LEMONADE framework is based on three main phases: *Pre-trip*, which focuses on teacher logistics and the procedures of organizing the groups and the activities’ contents; *Trips*, used by the learners outside the classroom using the LEMONADE mobile application; and *Post-trip*, which collates the trip activity results and the learners’ experiences by analyzing the collected media material gathered during the trip.

For the first phase, the LEMONADE framework provides an authoring web-based environment that includes the ability to create new trips, split the activity into groups, assign locations and associated keywords, to later identify collected data from those locations, and finally a task-creation process to produce the field trip activity content.

For the second phase, which consists of the trip itself, learners are required open and login to the LEMONADE mobile application, where they can select a group, download its related content, view the task description, review already collected data, view their status, manage a checklist of the current tasks and reflect on the results of
tasks that were already done. During the activity, the mobile devices’ GPS component is used to tag relevant collected data to a specific geolocation and to automatically tag preconfigured locations keywords to a medium, which refers to the current location.

For the last phase of the field trip cycle, the LEMONADE framework provides a web-based search-and-view option under a platform that presents the field trip groups’ collected data, which can include images, audio- and text-based comments, and a map of the locations the data was collected from.

One of the advantages of using mobile applications that include location awareness features is that learners adopt them easily, while performing outdoor learning activities. But basing the system on a GPS network can also increase the possibility of technical issues that may occur when dealing with outdoors activities. Delays and interruptions in the public GPS network (Zydney & Warner, 2016), or inaccuracy in the GPS location coordinates (Ryan, Hinze & Buchanan, 2006), will impair the user experience (Zydney & Warner, 2016).

2.3 Location-based Games

Games are a natural way for preparing people to acquire physical, cognitive and social skills. Electronic games are no different and have common features: high interactivity, fun factors, rules, scoring or other competitive system (Lope, López Arcos, Medina-Medina, Paderewski & Gutiérrez-Vela, 2017). Location-based games take advantage of the player’s surroundings to enhance the game experience by creating a connection between physical objects in the environment and the game’s virtual world, which leads to a memorable and richer player experience. Location-based games are much more integrated in real life than traditional games, in a way that can provide new interactions and exploration options (Procyk, 2013), and extend the boundaries of the cyberspace into the real world (Behbahani & El-Nasr, 2011).

Such games are widely adopted and played on smartphones, which are a perfect platform thanks to their sensors, like GPS, gyroscopes and other sensor-related components (Jukka, Tommi & Elina, 2013) (Kurczak, Graham, Joly & Mandryk, 2011). Due to the presence of these sensors with the modern mobile devices’ communication abilities, smartphones provide various possibilities to connect players to their environment and to create context awareness, which provides a solid ground for different narratives, mainly by placing the game contents in physical locations which are related to the game story and gameplay (Naliuka, Carrigy, Paterson & Haahr, 2011).

Many location-based games strive to create a fun playground experience for the player, and this atmosphere results in a learning experience, mainly because the games include exploration of different locations and objects with the help of media, sharing information and communication with other players (Avouris & Yiannoutsou, 2012).

A study by Zydney and Warner (2016), which explored mobile apps for science learning, indicated that the most common apps for the purpose were location-based. One of the examples they give is of an app Called MapHit that uses GPS data to track the students’ location during field trips, in order to save observation notes. The researchers also noted that location-based games may fall under the definition of immersive participatory, which they describe as a game that requires the players to search for clues using their mobile phone to solve a location-related problem. Another
popular app type that the researchers mention is location-aware games, which take advantage of mobile devices in order to detect the users’ locations, to provide information and clues for puzzle solving or handling a specific problem.

A story-based game called Backseat Playground was designed to solve a virtual, narrative-rich crime scene by moving to different locations that the game leads the player to. During the game, the virtual characters refer to real geo-related elements in order to connect the context of the game to the physical location (Bichard, Brunnberg, Combetto, Gustafsson & Juhlin, 2006).

Location detection is not the only strength of location-based games. Social interactions and relations also have a significant role in this kind of games, especially because they are closely tied to the usage of mobile devices that have many features and tools that enable and encourage different real-time communication (Jukka, Tommi & Elina, 2013). The social aspects of the games, particularly contextual elements and their social impact on the players, are taken into consideration by designers of location-based games at all development stages – from the idea to the narrative, game objectives, characters and related content (Diamantaki, Rizopoulos, Charitos & Tsianos, 2011).

Researchers point to several problematic issues concerning location-based games. One of these issues is the fact that the players need to look at the mobile device’s screen frequently, while they are immersed in the game’s virtual world. This may lead to physical harm, as the player may trip over or bump into different obstacles (Kurczak et al., 2011). Another possible risk is revealing personal or sensitive information about the player by sharing his physical location online, which is mandatory in some of the location-based games (Behbahani & El-Nasr, 2011).

There are challenges when designing and participating in such activities. These include a possible sense of isolation on the students’ side; issues with organization and the responsibility of the students toward their tasks in the absence of an instructor; and lack of control on the instructor’s side (Markova, Glazkova & Zaborova, 2017). Yet despite these issues, mobile learning and online learning in general have many advantages that learners and educators alike can benefit from.

2.4 Real-time communication for computer based learning

In the classic classroom, the teacher acts as the main source of knowledge, while he or she has the responsibility to teach the material to the learners, using suitable methods and communication skills (Zlatić, Bjekić, Marinković & Bojović, 2014). In order to perform this complex task, teachers use their personally acquired communication skills and style in order to receive and interpret messages, generate and communicate a response, and provide feedback (Duta, Panisoara & Panisoara, 2015).

There are many elements surrounding communication that can make it less effective, such as unsatisfying learning conditions, like a noisy classroom, physical distance between student and teacher, or cognitive issues that may arise from encountering unfamiliar concepts and words. One of the ways to overcome issues that may create communication barriers is to institute two-way communication, which focuses on
getting regular feedback from the learners in order to understand if the instructions were well received and well understood by the learner (Prozesky, 2000). Communication in the form of feedback plays an important role in learning and is a major part of the teacher-learner dialogue (Omoda-Onyait, Lubega & Maiga, 2013). It assists in keeping track of the learning process and reflecting it to the learner, helps to evaluate performance, and helps to improve later learning. Also, feedback is necessary to perfect the process and realize what future enhancements are needed for a specific learning activity (Popta, Kral, Camp, Martens & Simons, 2017).

Providing feedback and other communication elements in computer-based learning systems can be used in real time by the learners and educators in various ways via various tools. A real-time computer system was defined by Kopetz (2011) as a state “where the correctness of the system’s behavior depends not only on the logical results of the computations, but also on the physical time when these results are produced”. Real-time communication in computer-based learning refers to the use of online tools (such as video-based conferencing, live chat, live audio, VOIP and social networks) during a learning activity to actively affect its progress and outcomes (İşık, 2013).

In online learning, there are situations where physical contact among learners or with the instructor is missing. Tools for communication such as discussion forums, messaging and blogs are the way to encourage personal relations between participants such as students, teachers or teams, and they play an important role in keeping online learning humanized. Still, these types of tools are considered asynchronous and cannot be considered as real-time communication tools (Shahabadi & Uplane, 2015). “Synchronous e-learning is live, real-time (and usually scheduled), facilitated instruction and learning-oriented interaction” (Shahabadi et al., 2015). In contrast, asynchronous e-learning refers to a learning process which is not constrained by time or place, meaning that learning can happen anytime and anywhere, using online discussion forums and information sharing, with vast support for distant education (Shahabadi et al., 2015).

There are several tools for synchronous learning, the most popular being Skype. This tool can create personal and public interactions via instant messaging, conference video, files, screen sharing and live voice calls made possible by using VOIP technology (Hashemi & Azizinezhad, 2011). Skype can be used as a fast learning tool when fast communication between learners is needed, on deadline dates for example, all the while enabling giving and receiving clarifications and compensating for the learners being in different locations (Vinagre, 2016). Skype can also be used as an accurate and immediate tool for understanding real-time events, which promotes international relationships that step out of the regulated traditional learning process. In one case, students wanted to get information on political events in Egypt and initiated a Skype call to an Egyptian student to participate in a mutual exchange of information (Trust, Krutka & Carpenter, 2016).

The use of Skype and other similar apps, which are basically telephony software with additional service, was described by Hashemi & Azizinezhad (2011) as an education tool that can support and enhance the learning and represent the learner’s identity in the learning environment:

**Personal:** Students have a personal customized presentation screen for their comfort.

**Focus:** Critical for continuous flow and for the educators’ and students’ attention to be on topic.
**Privacy:** Students can keep their personal behavior unseen, especially due to the distance between participants and the teacher. This is not possible in the traditional classroom or other physical meeting places.

**Generated artifacts:** Such as text messages, shared images, videos, online links and audio can be shared.

**Multiple and parallel communication channels:** A text conversation can happen in private and public channels.

**Identification:** The participants’ names and details are revealed.

**Time extension:** As a tool to extend learning to other communication channels, which do not depend on the current session of communication, such as email groups, forums and social networks posts.

**Spontaneity:** Learners can meet instantly, without traveling, and easily share different types of content.

Location-based learning, in its essence, sends real-time students’ location from their mobile devices to a designated service that communicates related geographical data. Geo-collaboration is the performance of group-based activities that require the learners to interact with locations on a map that can be explored, and later to communicate via smartphones and like devices. Geo-collaboration involves collaborative exploration that refers to related content that requires social interactions and information sharing (Zurita & Baloian, 2013). For example, using real-time Google Maps indications under an integrated collaboration with the Google Places service can help enhance outdoors activities and act as an integral part of course design for educational purposes (Ermagun, Fan, Wolfson, Adomavicius & Das, 2017). A learning experience can be enriched by geolocation related data, for context awareness and communication among students (Avouris & Yiannoutsou, 2012).

Different services, applications and games use different methods to enable real-time communication between learners and instructors, but most of the communication occurs among the learners themselves (Jowallah, 2014). In educational location-based activities, such as games, the communication between the learners and the instructor is significant because of the physical disconnection between those two, which can lead to feelings of isolation and learner’s lower motivation (Markova et al., 2017). In some of these learning games, there are communication elements that are based on location sharing, messaging, and voice calls. But most of the time, these communication elements are not actually a part of the game platform itself, but rather act more as a framework for communication outside the learning activity (Israel, 2016). The Treasure-HIT platform offers its users a way to experience outdoor learning while using smartphones or tablets, thus creating context-aware learning contents, location-based tasks and enriched media content, and a sharing option based on the Facebook social network. Still the system lacks communication abilities in real time with the players during the game to monitor their action on the field and to communicate with them directly.

### 3 Introduction to The Treasure-HIT Platform

The Treasure-HIT platform is a web based platform for the creation of treasure hunt games that are played on mobile devices. Treasure-HIT was constructed based on two different components, an authoring web based platform, which provides all the options to create hunting game-based activities, and a mobile application that is used by the
player during game time to navigate to the designated game locations, handle the different challenges of the game, provide clues (Kohen-Vacs, Ronen, & Cohen, 2012) and share content on social networks (Israel, 2016).

The next sections will separately provide a description, technical details, workflow, technology usage and design implementations for each of the two system components.

3.1 Activities Editor
The activities editor (authoring platform), is a web application that includes the required features to create, save and deploy new or edited game activities. For each user, a set of unique authentication credentials are created, on request. After a short login process, the game-manager is presented with a games activity grid (Figure 1) and related “bread crumbs”, to create and manage games. The games grid includes a few UI controllers that presented for each activity:

- **Publish**: Enables the user to publish a selected game for a certain period of time;
- **Published Until**: Indicates the specific date range, from when till when, the game will be available for players to engage using the designated mobile application;
- **Game Name**: Customable game name that represents a specific game, which also be presented under the game activity UI on the mobile application during game time;
- **Game Code**: Unique numeric game code for the players to enter in the initialization of the game as their first step. The game code also acts as a unique ID on the system, for different usages, such as Database-related reference for the different game entities;
- **View**: Provides a preview for the selected activity and a simulation of a presentation under the game’s mobile application. Under the simulation session, the selected game is fully functional and can be used for testing purposes and refinements of the game’s structure and to test validity of the game flow and contents;
- **Results**: Includes two clickable buttons: a link to the game results page, which will appear for activities which got at least one action performed under a game session, and a second button that leads to the activity groups location map and instant messaging page, which are new features which were developed as part of the research for this thesis, and are described in detail over the next sections;
- **Gallery**: Presentation of shared images from the game activity which were generated on the players’ mobile device and uploaded to Facebook under Related Tasks;
- **Management**: Includes the option to print the activity station (QR codes), duplicate the activity and its data, share the game among other game-managers or delete game and related content.
To create a new game, the game-manager clicks on the “New game” link button, which directs to the game editing page (Figure 2). On this page, the game-manager can determine the game’s name, opening and ending messages, and a penalty interval, which is the time period which the player will has to wait, till he or she will be presented with the option to retry and re-answer and proceed. This page also includes route properties that are used to determine the order of the created station compared to other stations, which will be presented in game time, and a clickable button for editing and deleting a specific station.

When the user selects to create a new station, or edits an existing one, the system will navigate to the station editing page (Figure 3). On this page, the game-manager can determine the station’s name, the different tasks and types, questions and answers, clues, feedback and the verification method for the station, QR code or GPS location.
options. The QR code-based verification enables the game-manager to print the created QR code and place it in the game’s areas for the players to find. The GPS location verification method requires the players to navigate to the specific place, following directions given by the last station feedback, and check their location to verify they have reached the spot, using their mobile device. This last action sends the longitude and latitude coordinates to the system server, which responds with the appropriate message, depending on their success. The GPS location method is based on Google Maps API that enables the selection of a specific location or location radius on a Google map, by entering a specific street address using the Google Streets API or by using the map option to choose a location by clicking on a chosen location. This option uses Google Maps’ parameter to declare the required degree of accuracy for a specific station, according to a determined radius parameter value.

![Figure 3. Station properties editor](image)

3.2 Mobile Application

The Treasure-HIT mobile application acts as the coordinator between the game-manager created game content and the player. The mobile application goal is to accompany the player throughout the game stations, asking questions, providing orientation directions, clues, station location validation using a QR barcodes or GPS location-based coordinates, feedback for correct or wrong answers and different alerts for technical issues is appear.

As an initial step, first-time users are required to download the game application from the Apple App Store if they are using an iOS operating system based device or from Google Play if they are using an Android OS based device. After the download is completed, the user installs the application and supplies the required security permissions for the app, mostly to allow usages of the device sensors and collected data. After the app is opened, the player is presented with two options to search a game – by the user current location based on GPS coordinates, or by entering a game code.
After this pre-game step, which simulates a login procedure, the player will be directed to the game’s welcome screen that includes a welcome headline and a short game instructions text. After the user selects the game, the application directs the player to verify the current station location by GPS or scanning a QR barcode that the game game-manager placed in the location before the game (Figure 5). If the verification succeeds (a mandatory step for advancing to the station), a question will appear for the player to answer (Figure 6). After an answer is entered and verified, the user will be presented with a feedback alert window (Figure 7) to indicate of the correctness of the answer. If the player failed to provide a correct answer, the penalty period, accompanied by a wrong answer feedback message, will appear. If the player answered correctly, he or she will get clues for the next station, which usually include a set of directions for navigation (Figure 8). This work flow continues till the player reaches the final station, where the player receives a custom “end game” message (Figure 9).
3.3 Treasure-HIT Architecture and Technology

The Treasure-HIT platform is a holistic solution that consists of two main components, the Treasure-HIT authoring system and the Treasure-HIT mobile application, which are different in terms of function, structure, capabilities and purpose. The Treasure-HIT authoring platform is based on a web application that provides services to create new games and manage them, perform tracking over game play, depending on an authorization mechanism and relying on a Database for data storage, manipulation and modification.

The Treasure-HIT mobile application, however, provides the player with access to the saved authoring system games, serves as a mobile application driven by mobile gestures from the user to operate, does not rely directly on the system Database and is based on a game ID and not on the user’s personal authentication procedure and cookie for data pulling purposes.

For these matters, the two components developed under different tools based on different technologies, although, architecturally speaking, both applications, web and mobile, are based on an “Service orientated architecture”, whose realization is performed by a “Model View Controller” design pattern and framework, which will be discussed on the next sections.

3.3.1 Service Orientated Architecture

The Service orientated architecture (SOA) approach is adopted in different software to create loosely coupled services for the cause of distributed capabilities, which can later be operated by using a public external interface, that can be used to communicate with those services, with different external business clients (Kaczmarek & Węcel, 2008). This type of software architecture enables the mobility and modularity of different services that can be integrated and reused for other purposes than their original designated intention (Galster, 2010).

In Treasure-HIT, two applications were developed with an intention and understanding that the system will be scaled up by users and developers (Cohen, 2015). For that reason, the system was designed and developed under a focused effort of creating
different, separate services, that can easily be approached, tested and deployed, independently of other system components. The system components are based on an MVC design pattern, with the implementation of supported MVC frameworks, which help separating the different system components into conceptual entities. Both author and player environments adopted the principles of MVC, as the view controller is responsible for the UI presentation and user interactions, the controller is responsible to utilize the business login and to communicate with the model controller which is responsible for data related tasks, in this case mostly Database-related tasks (see Figure 11).

3.3.2 MVC Design

The MVC design pattern was used during the development process of the Treasure-HIT web and mobile application, for past and present features. The decision to use this design pattern under a .NET environment, is well supported and is one of the key values of SOA, which is separation of concerns. This is possible due to the idea that every component under MVC architecture is self-contained and deals with its inner logic under a well-defined responsibility (Freeman, 2013). The ASP.NET MVC and AngularJS frameworks also provide well-constructed communication capabilities, which both authoring and mobile applications require.

The Treasure-HIT authoring web platform uses the ASP.NET MVC web API capabilities which provide the option to create a web API with full support of the REST protocol (see Appendix A). Additionally, the ASP.NET MVC framework supports the ASP.NET that combines Razor annotations on the web page (see Appendix A) for the purposes of realizing different MVC view objects. This enables an easy and rapid development process based on templated web pages, using the template engine of the framework.

The mobile application is based on AngularJS framework, which also adopts MVC principles, to create separated application parts that can be maintained individually, and communicate with each other using regular JavaScript and AngularJS code references. The implementation of MVC under AngularJS is different than on ASP.NET MVC framework, mainly because it is the developer's responsibility to create a well-structured MVC design without “out of the box” built-in support. Angular allows this design pattern by providing an easy way to separate different components by their use, and later address them from different code snippets, but the option to mix view and controller functions is also available.
3.3.3 Treasure-HIT web application architecture

The web allocation is built upon Microsoft ASP.NET 4.5 under an MVC pattern 3.0 .NET framework, written in C#, and MSSQL for Database solution and an IIS web server for running and supporting the different application internal and external services usage. The use of the ASP.NET MVC framework, grants an easy option for the application to realize a web API based on the REST protocol. This API is used by the Treasure-HIT mobile application for different data manipulation usages and is also open for external users to use this API for their own mashup applications (Cohen, 2015).

An example for the usage of the system’s different components and related services is demonstrated by a description of the basic authoring workflow. The author enters the system by performing an authentication with the system server authoring Google OAUTH API. Then a new game is created using the system Database for storage and the MVC view for presentation. The controller receives a request from the view and commits the proper routing and communicates to the Database using the model, to fetch the relevant data for new created game and return it to the view. Then, the author creates a new station based on a QR barcode, requiring an API call to Google QR code services. After the station is finalized, the author decides to edit the same station and change the QR barcode-based validation to a GPS location validation type. This requires the system to communicate with the Google Maps API, to fetch relevant information and later store it into the system’s Database (Figure 12).
Figure 12. Sequence diagram of the Treasure-HIT authoring system, new station dataflow
3.3.4 Treasure-HIT mobile application architecture

The Treasure-HIT mobile application is built upon an AngularJS 1.0 Framework with Adobe PhoneGap 6.4 libraries. The application is designed for mobile device presentation and can operate under Android and iOS operating systems. The basic requirements of the application to operate under a mobile device are online internet connectivity, supported camera and a working GPS component.

The AngularJS framework is an open source project which is maintained by Google and based on JavaScript and a set of supportive liberties, which constitute the technology framework.

The AngularJS framework is based on an MVC design pattern (see Appendix A), which is a derivative from the MVC design pattern. Under this specific application, the Angular framework @scope references may be considered as View components, the model as the API class which communicates with the Treasure-HIT server-side Database and the different functions that warp each view as the supportive controller layer. The use of the Adobe PhoneGap libraries enables the developer to interact and perform actions using the device hardware, such as the camera, GPS components and audio speakers, and to create a more flexible responsive mobile application that will operate under different devices and operating systems with the same viewability properties and supportive functions. This framework makes it easy for the developer to adapt content and provides the possibility to focus on the development of the system functionality without major concerns about adopting and supporting different hardware and operating systems (Ruiz-Rico, Rubio-Sánchez, Tomás & Vicedo, 2013).

The mobile application is responsible for presenting and supporting the user with a chosen activity, interaction with the game’s server to post and get new data, use the device camera for QR barcode scanning and communicate with GPS satellites for location-based validations. An example of the usages components, services and device hardware usage is demonstrated by a description of a user interaction with application during over a new game, with a new created group and while facing the challenge of a location-based station. The player opens the Treasure-HIT application. After a splash screen, the player is prompted to enter a game code or use the GPS for nearby games. The player enters a game code, which sends a request to the back-end server is order to fetch the relevant game data based on the unique game code ID. After the data arrives the player is prompted to create a new group which is registered in the Database, by an API call, which includes a unique ID and the group’s name in its query. After this stage, the user is routed to the game’s welcome screen that is presented on the returned view page, and from there to the first station’s content. When the user arrives at the station, he or she performs a location-based validation using the mobile device’s GPS component, operated by PhoneGap’s native application API. After the player’s location is verified, an answer input is done using questions that are verified by a user generated “check answer” action. The sends an API request to the server API, which returns the appropriate clue to continue.
Figure 13. Sequence diagram of the Treasure-HIT Mobile application, Mobile device and backend components interaction.
4 Research Questions
Based on the review of the literature, and on the current state of the Treasure-HIT platform, I propose three interconnected research questions.

1. Which real-time communication facilities are required in order to enhance the learning activities conducted with Treasure-HIT?
2. How can we design and deploy real-time communication services supporting Treasure-HIT activities?
3. How do the new features incorporated in the Treasure-HIT contribute to the monitoring efforts of the game-manager and to the players’ experience?

5 Methodological Approach
This study is based on Design Based Research methodology, which is in vast use in the field of learning sciences, which TEL is a part of (Vanderhoven, Schellens, Vanderlinde & Valcke, 2016). Developments of new communication components and enhancements of existing ones were required during this study, processes which involved design, implementation, tests and evaluation. The Design Based Research methodology highly supports the ability to create initiated interventions for the researchers within the testing groups. This is valuable when challenging a process that is still under development, as well as committing refinements along the research process if outcomes are challenging the research hypothesis and the current development (Hoadley, 2004).

According to Anderson and Shattuck (2012), Design Based Research methodology consists of a several main concepts:

- Research can be conducted in a real educational environment, which allows the researcher to achieve a higher validity of practice and results in a naturalistic setting (Reeves, 2006), while considering the option of uncertain control over the research environment in-hand and its properties (Hoadley, 2004).
- Initiated interventions during the research process, that can include involvement in the educational activities and educational environment, involvement in the internal communications of the research’s participants, as well as knowledge sharing concerning the research progress and results (Barab, 2014).
- Using mixed research methods, such as field observations, interviews, questioners and numeric data collection under qualitative and quantitative approaches, provides the researcher with the option to implement, use and adapt the most appropriate research method for a dynamic situation.
- Multiple iterations during development, testing and reflection processes involving designing and testing different prototypes, with results that reflect enhanced and adaptive developments provide more flexible research environments. They include improvements and subsequent evaluations, with a general goal to better understand how the adjustments after each interaction influence the learning and practice of the participants (Barab & Squire, 2004).
- A collaborative relationship between researchers and participants helps to fill in possible knowledge gaps on the researcher’s side regarding the culture, theologies and politics of the specific test groups.
This study aims to use the above principles as guidelines and solid infrastructure for four main process that as defined by Reeves (2006), later expanded by Vanderhoven (Vanderhoven et al., 2016). I will analyze the communication requirements and gaps of the current system in order to support them, design a solution, implement and test the developments, and evaluate and reflect on the test results. These principles are organized in different steps which were implemented during the research and act as an indication of the different development phases and evaluations.

[Figure 14. Design based research workflow]

6. Development and Evaluation Plan

The development and evaluation processes were divided into four phases, following the four main design-based research processes as defined by Reeves (2006).

- Phase 1: Research and analysis of practical problems. In order to understand better the different user’s needs, problems and suggestion for improvements, a preliminary research was conducted on the potential users and on key experienced users. Additionally, part of the main motivation was derived from earlier studies that indicated future directions, for example, the need for a real-time communication between game-manager and players during game time (Israel, 2016).

- Phase 2: Development of solutions. Based on preliminary research and evaluation from past research, solutions were developed to act as a foreground to the current research efforts. Different additions to the developed solution were made after every interaction to enhance response to the new research data and analysis. Because the nature of development involves incorporating new features into an already existing and functioning system, it was required to encounter technical issues and limitations on the first stages of development. For that, a skeleton programming method was used. This method is based on creating “ready to run” applications without refinements and necessary requirements, for the developers to find and handle issues and for the users to experience (Katz, Merzky, Zhang & Jha, 2015).

- Phase 3: Testing, evaluation research and refinements. After each development interaction, a data collecting process was initiated, which was also carried out with semi-structured interviews with the study participants. This interview approach allows to conduct a formal interview, with a set of questions related to the issue and performed in a way that allows the interviewee to speak freely about other issues that arise from the questions if he sees fit (Cohen & Crabtree, 2006). After this process, the information was
analyzed and future enhancements and adjustments were suggested for the next stages of development.

- Phase 4: Reflection to produce design principles. Finalizing and summarizing the final research results in order to produce a practical design pattern that will use further developments and refinements (Reeves, 2000).

7 Implementation

7.1 Phase 1 – Research and Analysis of Practical Problems

7.1.1 Basic Perceptions
In late 2015, a dedicated Treasure-HIT activity took place at the Holon Institute of Technology, as part of an orientation meeting for new students. During the activity, there were 32 players and three game managers. The players included first year students in the Department of Learning Technologies who arrived at the pre-semester meeting. The instructors included lecturers in the Department of Learning Technologies who teach computerized design and computer science related topics. The age of the player's participants was between 21 and 30. The players were divided into eight groups of three participants each. The total time of the game was 30 minutes and was spread over more than 50% of the total campus area. Throughout the course of activity there were observations, which were recorded alongside comments and impressions that were taken from the game-managers’ general impressions.

At the end of the activity, semi-structured interviews took place with all the game-managers. Additionally, a field observation monitored the different groups’ behavior and actions during the game. The questions were focused on the different use cases that the game-manager performed during the game in order to get an idea of the different groups’ operations, technical issues that occurred during the monitoring and general impressions of the system and additions that they believed were necessary. At the end of the game session, a few semi-structured interviews took place with selected players as well. The interview questions focused mainly on the flow of the game, pedagogical and technical issues and general impressions of the mobile application and game contents presented in the activity.

After summarizing all the comments, observations findings and interview answers, I divided the key points into two groups that represent the main issues of the two user types. The game-managers mainly indicated the lack of monitoring options for the groups progression on the field, during game time. Additionally, both game-managers and player groups had indicated the lack of communication tools which realize interaction between the different entities during game time. According to the two groups opinions, this sort of communication option, will enhance the game flow, contribute to the general feeling and approach while handling different issues that may appear during game time, such as GPS issues or pedagogical challenges.
7.1.2 Interviews with Key Authoring Expert Users

Today, the Treasure-HIT platform is being used widely by various game-managers around the world. These users can be divided into two main groups: Users who only created a small number of games, and expert users who have created and published a vast number of games.

Interviews were conducted with all three of the activity game-managers, who are defined as expert users and took part in creating the related game activity. The game-managers were asked about their experience of monitoring different game groups during game time, about their preferred way of monitoring procedures, how they analyze game results, what they feel is lacking from the current system and what are their suggestions for future improvements. All the interviews were performed in person and took an average of 15 minutes to conduct. During the interviews sessions, handwritten notes were taken, which later were transferred to a computer, in order to save a precise copy of what was said. After the interview, I analyzed the written notes, and found some key issues which were repeatedly mentioned by the users.

1. **Lack of a monitoring ability for play’s groups.**

   Currently under the station editor, the Treasure-HIT system allows game-managers to create activities indoors and outdoors, by selecting a point on an interactive map or by specifically indicating a location through Google that provides the coordinates automatically. The ability to locate groups in alignment with game stations, approximately or accurately, is currently missing from the system. As was raised by the game-manager this information can contribute greatly to locate groups that had orientation issues during the game, including a possible indication of technical problems with the group devices and other difficulties of various kinds.

2. **Lack of communication between the game-managers and the groups during game time.**

   This issue became relevant in several situations:
   - Inability of groups to reach a certain location - this result was indicated using the system record in the database for wrong location indication of GPS based stations;
   - There is no help on the mobile help for various issues that could appear during game time. Such help could assist on orientation and content related issues with specific guidance;
• Technical malfunction related to location by GPS, without the awareness of the game-managers;
• Physical technical related issues that could be directly connected to Mobile device battery power, device GPS not working, physical inability of the group to perform a certain task, etc.

### 7.1.2.1 Field Observation

During the game I accompanied two groups, in turn, to observe their behavior in real-time, using the application, coping with challenges, technical problems, orientation issues and the challenges of the questions that were included in the specific game.

During the observation, I took handwritten notes, pictures and conducted conversations with the players. I was present in each group for about 10 minutes, one from the very beginning of the activity and the other during the final three stations till the end of the game. Each group had to pass through eight stations, which were introduced to each group in different order. The participants used their personal smartphones, iOS and Android types.

At the end of the game, a short interview was conducted with two participants of each game that I personally observed and one participant who was not observed by me in real time. Each interview took about five minutes. During the interview, handwritten notes were taken. Later, I analyzed my notes from the personal interviews, and found some points that I personally observed and some issues that were raised by the players themselves.

**Lack of proper indication for technical issues**

During the game activity, a few groups encountered GPS issues with their mobile devices. After several attempts to use the “check location” option on the Treasure-HIT mobile application, the group members found that the GPS option was not turned on the mobile device itself.

A second issue was the fact that there are no hints on location problems, such as wrong location, error responses came from the main server to app requests even after the GPS option was turned on and every parameter in the app and device seemed to work well.

**Issues with solving specific questions/puzzles**

Some teams found it difficult to resolve the questions posed on certain locations. It was mentioned that there were no new hints that could help to answer the question or any other way to bypass the specific problematic question to advance to the next station.

**Lack of human interactions between the group and the game-managers during game play**

Throughout different parts of the game, the players indicated that the general flow of the game could have been affected possibly by a direct live connection to the game-managers. When asked directly on what points of the game a human interaction is required, the players mentioned subjects which are concentrated on difficult questions solving help, questions formulation issues and a proper indication on the general game status, relating group’s status relativity to other groups (rank).
7.2 Phases 2 and 3 - Design, Development and Testing

**Proposed solution**

Based on the preliminary research findings, the research will address the issues raised most often by the game-managers and the players jointly. These issues are a common concern to both user types and can often be combined into a one methodological, technological and engineering solution:

1. Groups’ locations presentation on a map – groups will be presented on a map, associated by relevant location information for each activity, alignment with game stations, stations that have been reached and those whose locations were tested.
2. Automatic messages for players – messages will be sent automatically on certain events that will be triggered by certain registered data on the system.
3. Manual real-time messages – provide the ability to send custom messages to each group individually by the game-managers during game time.

The new proposed features will provide a solid communication layer between the game-managers and the players, which will provide a dynamic data mining tool for game-managers to better understand the game setup, the real-time game flow and players' different behaviors around the game field, and the quality of the existing challenges on the game. These new analytic data can contribute to various enhancements that can take place in current and future game activities, as well as to supply a solid ground for future research based on these records. The players will be proposed with the ability to better perform in real time by encountering less frustrations related issues, associated with the mobile game's application, system level factors and lack of help from the game game-managers.

**7.2.1 Real-time groups’ tracking map**

**7.2.1.1 Design**

The design phase began with a set of requirements, which were concluded during the preliminary research:

- Presentation of groups locations over a dynamic map
- Groups’ actions information for the specific locations on map’s markers
- Map filters for game stations and group locations points

These requirements reflect the need to present different real-time collected data to the game's game-managers, under a new graphical layer on the system management platform. To fulfill this requirement, it was decided to develop a group location system based on a dynamic map, which will reflect all the selected game group navigations experiences, interactions with the game stations, their data and interaction summary. The relevant data will be presented by a set of map markers with associated information windows, that will include an additional set of filters and in-map graphical indicators, to enhance the collected data analysis and to achieve a solid viewability of the selected group’s actions in real-time.

To provide a visual display layer for the groups data, it was decided to present the new Google Maps-based object on a new system page within the editor system. The access
to the new map feature will be available by navigation from the user’s activities grid, which will direct to the groups maps page. The new navigation button (Figure 16) will appear only for activities, which relevant group data is available in the database (at least one group activity record exists).

![Show groups' activities map navigation button](image)

**Figure 16. Groups’ activities map navigation button**

The game group location map page will include three main sections: a group selection drop down menu, which will enable to choose a group to be presented over the map, a map for the specific activity that will include the activity-related stations and a set of filters, in the form of checkboxes, which will act to present the user with the options to show/un-show activity stations.

The map’s markers will be divided into two types, which will represent a game station and a game group activity location point. These two different data types will be presented with a different marker icons (Figure 17) and different information details, which will show under the information windows when hovered over. The station information window will include the station’s name, the questions associated with the station and the type of the station (GPS/Barcode).

![Stations data types in-map presentation](image)

**Figure 17. Stations data types in-map presentation**

The station interaction information window will include the associated name of the station that the group is trying to interact with, the question itself, the interaction type, the answer that was chosen (in a case of a multiple-choice question) or written by the player and the final answer status and answer (wrong location, wrong answer and correct answer). To enhance the creation process of the new feature, in the design of its graphical user interface layout it was decided to create a basic mockup (Figure 14), that will determine the desired position and general viewability of each element on the page respectively to the general template view design of the Treasure-HIT authoring system pages.
7.2.1.2 Development (technical aspects)

The group location-based map feature required the combination of both the authoring system and the mobile game application. The authoring system is based on an IIS web server which runs an ASP.NET MVC design pattern-based application. This application is written on C# .NET, which handles the backend processing, and ASP .NET combined with HTML5 and JavaScript, for the front-end presentation layer, client calculations and processing.

The Mobile application is based on Angular 1.2 with PhoneGap JavaScript libraries, to enable utilizations of mobile device specific functions such as touch screen actions, camera usability, responsive layouts, support for different mobile operating systems, etc.

The game group location map feature required changes and new additions to the code which affected a vast amount of already existing system components. To describe the general development of this feature, each component will be described separately, as well as the relationship shared with the rest of the system. A general flow chart will describe the process flow of data between the various parts.

**Game Activities Grid**

To enable navigation from the editor's game actives grid required a new button under the results column of each game activity. The user can now click on it to be navigated to the new map page, displayed in the next section.

The new navigation button was added under the game grid view (Index.cshtml). The button was implemented with the qualities of responding to a click action by triggering
a new routing as a response. The routing action used an ASP.NET MVC Action Link annotation, which contains a controller routing path and the parameter values required for the map view to be initialized. The routing action uses the REST protocol HTTP request that is wrapped in the MVC framework.

```csharp
@Html.ActionLink(" ", "DynamicMap", "Map", new { selectedGroupId = act.ID })
```

Figure 20. Routing action to the Dynamic Map controller

**Game group location map**

To handle the routing request and data that the Action link contains required a new controller (MapController.cs), view (DynamicMap.cshtml) and View model (GameGroupData.cs).

The new controller class is responsible for receiving requests and responding with the designated view for the map interface, by encapsulating a few supporting methods, which handle the http request from the view, communicate with the Database model and store data in the view model for the view to use. The controller method addressed to initialize the map view by the navigation button is DynamicMap.

```csharp
public ActionResult DynamicMap(string selectedGroupId = null)
```

Figure 18. DynamicMap controller method

The main purpose of this method is to respond to the client with a view that contains the data required to be present under its user interface objects. To enable the use of the data that is transferred to the controller, it was required to create a new View Model (GameGroupData.cs). This class is designed to store information that will later be available for the view to use, by simple API based calls under the view based ASP page. The new View model includes a game group list field and a public method to point to this list.

```csharp
namespace MTH.ViewModels
{
    public class GameGroupData
    {
        public IEnumerable<SelectListItem> GameGroupList { set; get; }

        public GameGroupData()
        {
            GameGroupList - new List<SelectListItem>();
        }
    }
}
```

Figure 21. GameGroupData ViewModel

To populate the game groups data in the View Model, a call is made to a new method, which queries the Database model to get all the required groups data by a group ID.
public List<SelectListItem> GetGroupsNamesAndIds(string selectedGroupId)
{
    if (Enumerable_GAMEGROUPS().GroupsByActivityId
    (Convert.ToInt64(selectedGroupId)))

    foreach (var group in groupsByActivity)
    {
        SelectListItem groupObject =
            new SelectListItem { Value = group.ID.ToString(),
            Text = group.GroupName; }
        groupDetailsList.Add(groupObject);
    }
    return groupDetailsList;
}

Figure 22. GetGroupsNameAndIds controller method

To populate the game map object and display the selected game stations on the map, in
conjunction with the page loading process, a use of the *viewbag* dynamic view data
dictionary was implemented. The main quality of this property is its availability for
use, under the *controller* class and the *view*, under the same key name.

The *ViewBag* in this case is populated by a JSON object, which is responded to from
the CreateActivityPointsJson() method. The last step is using a Database model to
query and fetch the activity points from the Database, create a new JSON object and
return it as a string for the view to use.

private string CreateActivityPointsJson(string selectedGroupId)
{
    ArrayList groupsLatLng = GetActivityPoints(Convert.ToInt64(selectedGroupId));

    string convertedActivityPointsToJsonFormat = JsonConvert.SerializeObject
    (groupsLatLng, Formatting.Indented);
    return convertedActivityPointsToJsonFormat;
}

Figure 23. CreateActivityPointsJson controller method

The Game group location page is based on an ASP.NET MVC View object. The view
includes the map object and its accompanied user interface elements. The view page is
initialized after the controller response includes the view object back to the system
page template, alongside with the *ViewModel* object.

The *view* implements the Google Map JavaScript v.3 API, for map displays and
realization for different filters and presentation modes. For the initialization process of
the map, a use of the saved activity logs from the saved resource *Viewbag* which is
saved to a new points variable.

ViewBag.activitypoints = CreateActivityPointsJson(selectedGroupId);

var points = @Html.Raw(ViewBag.activitypoints);

Figure 21. Shared MVC View object, used in map for controller and view entities

Later, the points variable is used in the map object for the center parameter.
The points array is also used to present the activity’s points on the map, decide on their icon appearance and their associated information window. Each Activity station is present in a specific marker that describes its type, GPS or Barcode, based on available station details. A mouse hover action on the station map marker will open an information window, which presents the station name and its related questions.

The second main quality of the Map controller class is to respond with current group locations and related details. To show the group location, the user clicks on the Group selection dropdown menu, which presents the group’s name, and selects one of the available items for a group selection. The dropdown menu action triggers a new request to the controller, which returns the group locations and associated data to the view. To realize the selected group location request, a new Action link method was created with a specific routing.

```
[HttpPost]
public ActionResult Create(string groupId)
{
    string groupJson = CreateGroupPointsJson(Convert.ToInt64(groupId));
    return Content(groupJson);
}
```

The method is getting a create routing call from the view and return the groups’ data under a JSON formatted string.

To accommodate the group data object, a new method was created (GetGroupDataByGroupId). The method is responsible for querying the Database.
model, in order to fetch the group data by a given ID. The data itself includes the following fields: latitude, longitude, point accuracy value, point name, related activity latitude, related activity longitude, the related text and the answer text.

The details are presented in the group location information window, by a hovering user action, with a short summary that details the group activity in this specific location.

The summary is available by mapping all the relevant Database registrations that were made for a specific point, by comparing the latitude and longitude values with point IDs on an inner loop under the view class.

**Treasure-HIT mobile application development**

To enable the system to display the data of the separate groups over the map required registration of the data sets on the Database. The information about the group's activities comes from the user actions performed on the mobile game application. Some of the data registration processes which were required to register and later realize the various indication on the map, already existed on the application and in the Database, but in order to fulfill the requirements of the map feature, it was required to perform further development of and additions to the mobile application, on the Treasure-HIT system and on the Database.

The data transformation between the mobile application and the Treasure-HIT system is based on API Ajax HTTP requests that are made from the mobile application to the Treasure-HIT API controller. Each addition of new registered data is performed on both the mobile application and the Treasure-HIT system in parallel, to enable both ends to send the data, receive it and respond appropriately.

The mobile application additions included longitude, latitude and point ID data, which is sent to the server by an API call and later saved to the Database under the GameGroupLog table.

```javascript
if (this_deviceSensors.checkConnection()) {
    this.showAPIRequest("/api/gameGameLog?myGroupID=" + this.myData.myGroup.ID + "&latitude=" + myLatitude + "&longitude=" + myLongitude + "&accuracy" + this_deviceSensors.thiAccuracy + "&action" + "&note" + "&datetime" + now.toString()) + 
    "& fflush" + myPoint, "GET", nil, "XML", function (data) { 
    log.Registered = true;
    }, false);
}
```

**Figure 27. Mobile application API call for group’s activity logs registration**
Encountered Technical issues and Limitations
Since one of the initial steps to define the technical status required code additions and enhancements, the complete Treasure-HIT echo system was reviewed by combining various reverse engineering procedures, debug actions and code reviews. During this initial research, two crucial issues were identified:

1. Group locations were registered to the DB with null values. The issue was fixed by adjusting the group's location registration to be based on the GPS sensor interaction from the user actions themselves and not basing on the

Figure 28. Sequence diagram of the Real-time groups’ tracking map feature backend data flow
logAction function sensor data fetching, which was not replying with a valid value.

2. There was no link between the location points of the group activity to the station on the Database. In order to supply the required information for the game group activity on the game field, it was needed to add a parent point ID registration to the GameGroupLogs Database table. The new data was added by using the current mobile application Angular framework $scope object, which stores the station ID for each encountered question, and sends it using an API call to the Treasure-HIT server side, which registers the station on the Database for the specific activity log.

### 7.2.1.3 Testing and Refinements

After the first development phase, the feature was tested under three internal field tests, committed by the developers and reviewed by three expert users. The technical tests had indicated the need for a few code changes, in order to enhance the performance of the new feature and its general user experience. Furthermore, the expert users review, had indicated the need for a few graphical improvements, which focused mainly on the visibility of the stations and group data presentation under the different markers' information windows.

In order to evaluate the development up to this point, the feature was reviewed with a few three keys expert users, who operated and managed a Treasure-HIT game and also acted as players. Additionally, the feature was presented, tested and reviewed by software engineers, quality assurance engineers and user interface experts who reviewed the system for the first time.

The review summaries for both groups indicated two additions that would improve the authors’ map orientation and introduce more informative research tools that could be used for game progress monitoring while operating and managing an activity in real-time:

1. Presentation of the groups' trajectory of the on the map in scheduled order during game time
2. Display link between the different groups activity points during the game, depending on the various activities which are displayed by the game stations.

Those two additions led to the development of two additions to the current feature:

1. Presentation of the group's estimated path along the different group activity points that were committed during game time, in ascending order.
2. Presentation of graphical in-map lines drawn to create a connection between the group activity points and the relevant parent station.

The first addition that was developed to address the required addition was the group activity tracking rally points. In order to display the new tracking rally points, it was decided to add a new checkbox on the map's side bar. By checking it, the user sees in-map rally lines that are drawn between every activity point node, with associated arrows pointing in the direction of the group’s physical progress and supplying an indication of the group’s movement, arranged by the time the action was done.

To create and place the different rally lines between the activity points, it was required to create a new Google Maps PolyLine type array under the map's view class. The
array itself was added by a dedicated method that is called by a JavaScript function and triggered by a checkbox event. The method operates on the group current points JSON string object, to add every group's points into the PolyLine array with associated display properties. The rally points PolyLine can then be removed from the map by un-checking the Show estimated group path checkbox.

```javascript
polylinesArray.push(new google.maps.Polyline({
  path: err,
  strokeColor: checkpoints[i][0][1],
  strokeOpacity: 2,
  strokeWeight: 3,
  fillColor: checkpoints[i][0][1],
  fillOpacity: 2,
  icons: [{
    repeat: '70px',
    icon: iconsetings,
    offset: '100%
  }]
})
```

**Figure 29. PolyLines data entry for group estimated path creation**

![Map with PolyLine data entry](image)

**Figure 30. Group Activity points estimated path display**

The second addition that was developed to address the new requirements is the new display option for presenting the relationship between a group action and the station the action was addressed to. For this feature, it was decided that drawing direct lines from the activity points to the parent station should accomplish this. This graphic indication, is scheduled to address the additional purpose of accumulating data, for the benefit of understanding the nature of the station and possible technical issues related to the station, in such aspects as possible location problems, mobile device-related problems, group orientation difficulties or pedagogical-related aspects, such as the nature of the formulation of the questions and clues.

To address the development of this feature, it was required to first create a new method that will handle the process of connecting between the group activity points and the parent station. To perform filtering of the different groups activity points, it was decided to use a Map and Reduce programming model, which will map all the game's station points and then perform a reduce function that will summarize only the relevant station and their groups activity points into separate groups under one object.
To implement the new model, a new section was added to the `GetGroupDataByGroupId` method under the Map controller class. On this method, a new `foreach` condition was assembled to summarize (reduce) all the relevant data of the selected group ID activity points (map), by comparing the station and group activity IDs. On each comparison action that returned a `true` operator, a new value is added to the `groupPointData` array, which is later served to the `view` object, by the group selection action link call-back.

```javascript
foreach (var element in getGroupLog)
{
    if (element == GetPointByID(element.PointId))
    {
        groupDataList.Add(groupPointData);
    }
}
```

**Figure 31. Map and Reduce inner loop for station and group activity points**

To present the connection points between the station and group activities on the map object, it was required to add a new function under the `view` class, which will be activated when the new dedicated checkbox is checked.

The new function performs a loop on the relevant returned data for the group's data, matching and collecting the relevant stations and points data into a new `PolyLine` type array, which is used by the Google Maps API to make the connections and draw the lines between the points over the map object.

```javascript
if (document.getElementById('allPointsFromChildToParent').checked)
{
    for (var i = 0; i < gameCheckPoints.length; i++)
    {
        var pointToStationPolylineArray = [
            { lat: parseFloat(gameCheckPoints[1][1][0]), lng: parseFloat(gameCheckPoints[1][1][1]) },
            { lat: parseFloat(gameCheckPoints[1][1][2]), lng: parseFloat(gameCheckPoints[1][1][3]) }
        ];
        var pointToStation = new google.maps.Polyline({
            path: pointToStationPolylineArray,
            geodesic: true,
            strokeColor: '#FF0000',
            strokeOpacity: 1.0,
            strokeWeight: 2
        });
        pointToStation.setMap(map);
    }
}
```

**Figure 32. Draw polylines on the map between stations and relevant activity points**
The second phase of development includes the inclusion of the relevant education-related content of the questions and answers that a group receives at a certain station. The information entities that were decided to be relevant for the info windows presentation and later research included the content of the questions, the answers and the parent station. The data fetched from the questionCtrl.js controller class must be saved on the logAction methods under the GeneralHandler.js factory class. To commit the new data, save action, a new section was created under the method, which is activated from the questions controller with the relevant data. The new data particles are then added to the logAction API call. On Treasure-HIT, new parameters were added to the API controller calls, in conjunction with new columns in the Database, to store the newly received data.

```
saveQuestionText: function(questionText) {
  myQuestionText = questionText;
},
```

```
saveAnswerText: function(answerText) {
  myAnswerText = answerText;
},
```

Figure 33. Group activity rally lines drawn to related station

Figure 34. New calls to generalHandler factory class methods
7.2.1.4 Field Observation

After the development of the Real-time groups tracking feature and refinement stage, a field observation and evaluation was carried out. In early 2017, an activity took place in the Holon Institute of Technology based on four different stations that were randomly ordered and performed by four players, who were divided into groups of two and monitored by one expert game-manager. The activity content was focused on navigation tasks that conducted movement between different buildings around the campus. Before the beginning of the activity, the participants were instructed to perform a few location check actions and make intentional mistakes in answering the questions. The instructions were to provide more data about the route of the groups and the question and answers under the activity points information window.

The activity took 15 minutes on average. During this period, the game-manager performed tracking actions using the different map features over groups, with focus on the registration of new points and the presented information for the groups’ actions under the information windows of both stations and activity points.

During game time the groups encountered two main issues. One was related to navigation issues which occurred because of the lack of information on where specific buildings are located. This created various indications of “location check” actions over the map. The second issue was the difficulty of the group to ascertain the location of a particular building because of low GPS reception, which caused location error indications over the map, with direct connection to the specific building (Figure 33). The group moved to different locations around this building until GPS reception picked up and allowed for proper location validation.

After the activity, semi-structured interviews took place with the activity’s game-manager and the players (Appendix B). The interview with the game-managers focused on his experience with the map tracking features, including questions about the real-time indication markers registration and appearance over the map, their usage and feedback on the rally point and the points and station related features, with the
possibility of a free response to get general impressions of the new features’ capabilities, and suggestions for improvements.

The game-managers testified that the real-time group tracking feature contributed in the effort to get a clear idea of the game flow and related group issues. But he also indicated that although the different issues can be monitored during the activity in real-time, there is no way to respond to the groups in order to provide assistance of any kind. On an educational notion, the game-managers also noted, the answers text which was presented under the different activity point information windows can act as a tool to monitor an education activity, and to better understand the learning challenges from the students’ side. Another feedback was focused on the difficulty of understanding a certain path for a group that made “check location” actions at close intervals (Figure 34). The game-managers noted that “after a while it was very hard to track the path of one group, the map rally points seemed to merge with one another in that way that made it hard to monitor the direction the group had gone and what the real source was of the issue of disorientation”.

The players’ interviews included questions related to their experience with the mobile application, with focus on the different issues that they encountered during the activity.

7.2.2 Game-manager and Player Instant Messaging service

7.2.2.1 Design

As it emerged from the preliminary research, the second new feature that was expected to contribute much to enhance the player's game and general game flow was the ability for the players and game-manager to hold real-time communication during game time. This feature requirement was also raised by the first field observation which took place after the implementation of the real-time group tracking map discussed earlier, and summarized in the need for a way to solve real-time issues with orientation, clues and questions clarifications during game time.

During the design process, three approaches were considered for implementing a real-time communication between the two parties. One approach was focused on taking advantage of the HTML5 sockets technology, which provides a way to maintain asynchronous communication between the server and the client, while using the .NET

![Figure 36. Registration of wrong location check points due to a GPS reception issue](image1)

![Figure 37. Registration of wrong location check points due to an orientation problem](image2)
SignalR library. The second approach was focused on using an unwrap REST protocol based communication to committed long-polling also based on the SignalR library, and the third approach suggested the usage of realization of requests which are pre-scheduled by different user actions.

The dilemma that was at stake focused on the ability to conduct real-time communication under the architecture of the current authoring system infrastructure and available resources (hardware) on which the system is based in order to operate. Additionally, the mobile application, which is based on AngularJS and PhoneGap frameworks, was also considered for the possibility of using open sockets, managing of those sockets and limitations that may appear on behalf of the frameworks and the mobile operating systems. Questions concerning the type of technology to realize the feature focused on two main issues:

Server Computing Power: Low computing power can cause major delays in the message service itself and affect the general user experience on the mobile application and authoring managing tools. Additionally, the system's web server has a limitation of the number of open sockets it can handle at a given time. This can result in limited availability, considering the messaging service and other system features.

PhoneGap framework: Possible limitations: the ability to transfer data considering restrictive technical issues that may exist in PhoneGap combined with the SignalR technology, in current and later PhoneGap versions.

SignalR library: The SignalR library also includes a possibility to use long polling procedures in order to carry out real-time communication, but is also considered too expensive with respect to the existing system resources.

After consideration of the current system architecture and the given risks, it was decided to realize the new communication layer using GET/POST requests, based on the HTTP REST protocol. To implement the user insertion of new message, it was decided to create a new TextBox client side component, under the map view class. The TextBox inserted text data will be sent to a new backend method which will register the message to the system Database for the specific selected group. The message itself will be present to the player under a new alert form, which will be triggered by a user action of answer check under the mobile application. The game-manager will be able to decide if to show the message under a current or wrong answer or to disable it completely for presentation. The message content itself will be presented to the player user by an alert form which will include the message's text.
7.2.2.2 Development (technical aspects)

The development of the game-manager/player messaging feature, is based on the implication of the game real-time group tracking feature dedicated view, with an expansion of the view and associated controller to support the insertion and storage of new message related data to the system Database. Another aspect of the new feature is the adjustments required for the Treasure-HIT mobile application to request and get the message data when required and to present it under a new alert form.

**Message text input**

To enable the user to insert a new message for the specific selected group, it was required to insert a new Textbox under the DynamicMap.cshtml view class. The Textbox was implemented using an HTML form input tag with an associated submit button. The inserted text input is sent to a new JavaScript function that activates an AJAX call to a new controller method under the MapController.cs, that receives the data and uses a model based method to store it in the Database.

```javascript
function writeGroupMessageToDB(message) {
    var groupid = 'points';
    console.log('id = ' + groupid);

    $.ajax({
        type: 'POST',
        url: '@Url.Action("MessageHandle", "Map")',
        data: { "groupId": groupid, "messageText": message },
        success: function (response) {
            alert('The message was registered and will in the next answer check action, for the selected group');
        }
    });
}
```

**Figure 39. New message data encapsulated JavaScript function**

In order to store the new message data, it was decided not to use any of the existing Database tables, but to create a new table under the name of MessagesData. The new table includes the columns: “ID” to keep an ordinal order of the rows, “messageText” to store the actual sent message from the new input textbox, “GroupId”, which fetches on using the selected group ID from the map groups dropdown and “Enable”, which is a Boolean field that enables the game-manager to decide if the message display should be turned on or off for display under the mobile application.
To enable the mobile application to get the stored message text form the Database, it was required to create a newApiController class, which will respond to the mobile application request with the proper message text, for the group that requested it.

To realize the new API call, a new GET request method was created under the MessagesController.cs. The new method gets the GroupId sent from the mobile application API request, and responds with the relevant message text fetched from the Database for the specific group ID. It is important to mention that under this development stage it was decided to get and present only the last record that was entered to Database by the game-manager.

```csharp
// GET api/Messages/5
public HttpResponseMessage GetMessageForGroupId(int GameGroupId)
{
    IEnumerable<MessagesData> groupMessages = db.MessagesData.Where(c => c.GameGroupId == GameGroupId);
    return Request.CreateResponse(HttpStatusCode.OK, groupMessages.Last().MessageText);
}
```

Figure 40. API call for the selected group message text data

**Treasure-HIT mobile application message alert form**

To enable the message content to appear under the mobile application, it was decided to create a new API requires to the Treasure-HIT server, which will be sent after a mobile application user makes a "check answer" action. The new request to the server was accomplished by creating a new API requiring the APIAction class, and a new API function under the checkQuestion method on the questionCtrl.js class.

```javascript
generalHandler.shared.APIAction("/API/Messages?GameGroupId=");
$scope.generalHandler.appData[thisGroup.ID, "GET", null, "12345", function (data) {
    $scope.popUp.setPopUp("Message from supervisor",
data,
    "Close Message", "message", function () {
    });
}).false);
```

Figure 41. API call to get text message from backend

The decision to implement and present the response under the "check question" user action was made based on a statistical calculation, based on a snapshot of 1,548,247 log records from the Treasure-HIT production system’s Database. The snapshot data is based on logs that were registered from the beginning of 2014 till late 2016. The data were analyzed by dividing and summarizing the quantities of the players’ actions by type (Figure 39). The summarized data indicated that the most common log registry, which is related to a user action among participants of the game, was the right and wrong answers log registries. Those logs were registered to the Database by a check answer user action in the mobile application.
The message content is presented under a custom-made alert popup with a designated new image to represent the message type alert, a header which indicates that the message came from the game-manager and a message text. The alert is dismissed by pressing the “Close message” button that enables the user to proceed to the next game action.

7.2.2.3 Testing and Refinements
After this development stage, tests of the new addition were carried out by a few stakeholders, who included two experienced game-managers and two players. The tests included sending messages in various configurations, such as long texts, characters in different languages, and using special characters and attempts to
challenge the system by stress tests, by sending multiple messages at short intervals and from many users at once.

The application tests included viewability tests for the appearance of the text inside the message text box and occurrence of the message under the expected players’ events. Following the tests a few conclusions emerged which resulted in some required enhancements. It was decided to change the message indication in the mobile application, not only for the player action of “check answer”, but also to append it to a check location action. The new message API call was inserted both for GPS and QR barcode validations under the $scope.checkLocation Angular function (Figure 38). This came after a game-manager tested the feature by sending a message to a certain group about a location precision issue, which was seen by the player only after the location was found and the answer was answered, which was not the game-manager’s original intention. For the instant message feature under the authoring platform, it was decided to change the message box from a simple textbox to a text area type textbox, with a specific width and height, to allow for the appearance of the complete text.

```javascript
$scope.checkLocation = function () {
  $scope.generalHandler.saveAnswerText($scope.answersString);
  generalHandler.shared.APIAction("/API/Messages?GroupId=" +
  $scope.generalHandler.appData.thisGroup.ID, "GET", null, "12345", function (data) {
    $scope.popup yp.setPopupYp( "Message from supervisor", data, "Close Message", "message", function () {} )
  },false);
}
```

Figure 44. Check location with get message API call

**7.2.2.4 Field Observation**

After the first development phase, a field observation and evaluation was carried out. In early 2017, another activity was held at the Holon Institute of Technology.

Prior to the activity, a group of two expert Treasure-HIT game-managers created an original activity that included four stations in total, which will be presented in mixed order. The activity focused on orientation around the campus between different service suppliers. The activity game-managers under this specific experiment were instructed to provide two stations that should include misleading instruction for navigation to reach specific selected stations based on location validation. This type of instruction came in order to simulate a situation where a location validation is recorded on the Treasure-HIT Database as a location error. The error indications will appear on the groups’ location map feature and will notify the game game-managers to send a custom message to the selected group, to provide new navigation orders or clues.

Prior to the activity, the activity participants downloaded the mobile application to their phones (update if already installed). After this step, the participants were instructed as to the activity goals and the current research efforts and focus. The activity itself took place around the college campus and included six participants in groups of two. The participants were students from the “Learning Technologies Department”, between the ages of 22-40.
The game activity took an average of 20 minutes. During this period, the game-managers performed tracking efforts, using the “Group location map feature”, which found the problematic stations indications that were indicated by the groups’ “check location” actions. In response, the game-managers sent appropriate messages to direct the participants to the correct location. During the activity, the game-managers also found and reported a possible issue with one of the group’s mobile device’s GPS component, indicated by long intervals of new error records in the Database from a location check action. Additionally, a long time elapsed between registrations. The game-managers sent a specific message to the specific group and the issue was solved shortly after (the station was not one of the deliberately false instruction clues).

Figure 45. Player Indications of the progress of the group and the provision of an estimated route on the map

Figure 46. Game-manager, while monitoring a group progress in real time

Figure 47. game-manager message displayed over a player’s smartphone

After the activity, two different online questionnaires were sent, one to the activity’s game-managers and one to the participants (Appendix C). The questionnaire included open free textboxes and multiple choice questions about the activities on technical matters, specific message feature impressions, general graphical UI usability and general satisfaction with the Instant Message feature with addition of a free text option for any type of remarks and suggestions. Semi-structured interviews (Appendix B) also took place with the activity’s game-managers and a few chosen players. The interviews focused on their personal experience with the “instant message” feature and the reaction of the groups in the field.

The students’ feedback indicated a general satisfaction with the instant message feature, but with a few exceptions. One open text feedback indicated that the message
appeared more than once, even after the group had advanced to the next station. Another open text feedback indicated that the message appears after the group achieved correct location validation. Another open text feedback noted that messages should appear not just for correct and wrong answer check actions, but as an instant message, where the message is sent and appears after a short while on the device application without a specific relation to the user action.

The game-managers testified that the instant message feature seemed to help the group’s navigation efforts and was an effective way for communication with the game groups during the game. Along with the generally positive feedback, the game-managers had a few constructive remarks. One of the game-manager noted that “there should be a character limitation for message insertion. It is not clear how long the message can be and show properly on the mobile device”. A second remark from the same game-manager noted that “It’s not clear when and if the players did receive the message, and if they receive the message, do they read it?”.

**Further Requirements and Refinements**

From the feedback from students and game-managers it was decided to enhance the current solution on both the instant message and mobile application.

For the first enhancement, it was decided to change the logic of how the message appears. The message appeared after every check answer question. The change in logic required that the message will appear only once for the player with additional indications that inform the game-manager that the message has been read.

After a short research, it appeared that both new additions can be handled under the same solution: A new “read” column was added to the messages table under the system Database. The new column is Boolean, whose “true” value is set to read messages by the player and the “false” is for unread messages. These indications enable the mobile application to query the data base using the same API call that already handles the get message. The request response included the “true/false” values. The show message function is added with an “if” that under a true value will show the message, and under false value will show nothing.

The true and false values are also indicated on the Treasure-HIT authoring platform, under the group’s map feature, on the message text box panel. The indication comes in a form of two images. “Check” image (Figure 46) appears on true value and “uncheck” image (Figure 45) appears under false values. The check indication and uncheck values are also indicated by the same “read” rubric which relates to the selected group’s ID value, by a new Database query.

![Figure 48. Read indication is set on false](image-url)
The game-managers feedback raised the concern of large text messages that may appear with missing text, under the mobile application message pop-up windows. To engage this issue, it was decided to add a character counter, appended to the text area textbox (Figure 48). The character count was limited to 120, to optimize the popup view and calculate by a new JavaScript function under the charLimitCalc() and a supportive call from the actual ASP.NET multiline textbox (Figure 47).

8 Discussion

During the development of the Treasure-HIT, two main evaluations took place, one after the first phase of the development of the real-time tracking group map feature and one after the first phase of development of the instant message service feature. The purpose of those two evaluations was to test the usability and technicalities of the two new features to encounter technical issues and to evaluate acceptance by the two user types, game-managers and players. The first evaluation was conducted only for the real-time group tracking map, as the instant message feature was in early development stages at the time, and the second one included expert system game-manager and students, and was conducted on both features. Both evaluations focused on the experience of the game-managers while using the new features, their usability and the behavior around them, technical issues and limitations, and finally suggestions for improvement from both the game-managers and the players.
In both evaluations, the game-managers expressed general satisfaction with the features, and a certain gratitude for the long-needed tracking functionality. Also, both game-managers and players indicated that the instant message feature was valuable for maintaining the connection between both user types, that it helped the of the game flow, and relieved the frustration previously felt without the real-time connection between the two entities.

Shahabadi & Uplane (2015) claimed that Synchronous e-learning should be conducted with tools that provide the ability to produce live analytical tools, chats and create communication with the learning activity instructor, who in his turn can provide the help and support the learners need. These recommendations are in accord with the development efforts of this research, especially in the sense of the real-time support provided by the game-meagers, while using the different new communication services. In addition, the new communication capabilities are consistent with the recommendations of the future developments mentioned in Israel (2016) study. Those stated the need to improve the communication abilities between the player and the game-manager, in order to ease frustration stemming from different issues during game time. These additions also ease the physical disconnection between the player and game-manager, which can lead to feelings of isolation on the learner’s part (Markova et al., 2017).

During the evaluations, some issues were raised by both the game-managers and players. The game-managers reported that the groups markers on the map were not clear enough to exclude them from other map related markers and icons, the tracking rally points drawn from the groups markers on the maps interfered with one another, and it was not always clear in what order the stations were, as after a while and especially when a group had many location-related errors, the rally points became massive which created difficulty to analyze the progress of the individual groups under real-time monitoring. Further feedback indicated need for bidirectional communication with the players, or at least a proper indication of whether the user had read, or not, the sent message from the game-manager. Additionally, the game-managers also asked for a way to make messages to more construed, as in fixed formulations, which can save time in typing a new message every time for each group, as well as avoiding the message sending action at all, by providing more in-app alerts for specific common events. For the game-manager, these additions can save critical time during the game flow, and contribute to a more fluid game for the players to experience.

The players reported that the instant messages helped them a lot in solving orientation related issues, as well as gaining a general feeling of connection to the game-manager. The players provided technical feedback that included remarks on the messages repeated appearance, after the issue which was indicated on the game-manager map was already solved, such as a poor GPS reception. The need for instant communication with the game-managers, in the players’ opinion, could save time on problematic tasks that can derive form pedagogical matters, such as missing information regarding the presented questions.

To conclude, real-time communication features were implemented on the Treasure-HIT ecosystem, to better analyze the game progress and make improvements on the game-manager side, and to provide the players with a better game experience, by strengthening the connection between the players and the game-managers, improving
the game assets by enhancing the context awareness using communication (Avouris, et al., 2012) and to meet various needs that arise while playing in real time.

The real-time communication additions that were added on the authoring side added a critical value that contributed to the game-manager during and after game time. The game-manager made extensive use of the different monitoring features during the evaluation, in order to get a better idea of the game flow, to evaluate the players’ results (Giemza, et al., 2010) and also to move towards closure for the moment, leaving room for future enhancements to existing and future activities, pedagogically and geo-learning related.

The players were introduced to the instant message feature only in the second development phase. The contribution of the instant message feature translated to satisfied feedback reports from the players, which indicated a stronger sense of mutual communication between the game-manager and the groups in the field. Furthermore, the feature helped to solve real-time technical issues, which saved the players time and frustration.

9 Summary and Conclusions

The research conducted in the thesis studied the addition of real-time communication features to the Treasure-HIT platforms, both on the web authoring platform and on the players’ mobile application, to better analyze and control the game flow of the players, to better understand the effect and relevance of the educational material presented in an activity and to catch the loose ends, and allow future game and learning material enhancements and refinements.

In Section 4 above I proposed three research questions:
1. Which real-time communication facilities are required in order to enhance the learning activities conducted with Treasure-HIT?
2. How can we design and deploy real-time communication services supporting Treasure-HIT activities?
3. How do the new features incorporated in the Treasure-HIT contribute to the monitoring efforts of the game manager and to the players’ experience?

The first research question examined which real-time communication facilities are required in order to enhance the learning activities conducted with the Treasure-HIT platform. This research examined the required services to create real-time communication between the game manager and players, during and after a game activity. To answer the question, efforts were focused on finding which new real-time communication layer would benefit Treasure-HIT users most, what data should be presented and saved, what type of communication should take place between the player and the game-manager and how this real-time communication can contribute to the educational purposes of the learning activity. As a first step, preliminary research was conducted. The research consisted of a field observation of a few experienced game managers and students, during an ongoing game activity. During this activity, notes were taken, and it was followed by semi-structured interviews both with the game-manager and with the players. The conclusions of the field observation and interviews led to the identification of several main needs of game-managers and
players. The needs focused on two primary features to be developed, a real-time group location map that will present the players’ progression and actions on the game field, and an instant messaging service, that will allow the game managers to send real-time instant messages to the players.

The second research question examined the possible ways to design, implement and deploy real-time communication services that will integrate with the existing Treasure-HIT activities and its ecosystem. The intention of these new features was to support real-time communication that will answer the needs of both stakeholders. Each feature was designed and developed separately to examine relevancy, effect, integration and testing efforts with the other system components. Each feature included a requirements list which followed a design process that considered the technical efforts that should be made, the effect on other components of the system, and user interactions, all with consideration of the user interface design and experience. The development itself was based on iterations: first, the feature was developed and tested for basic validations, then a series of fixes and improvements took place both in the code, mainly for efficacy and system considerations, and for the visual appearance of the features. The second stage included a field test that was conducted by real users, who summarized the experiment and the tests results, which led to new requirements and enhancements, followed by a second round of development in order to meet those requirements. After the implementation of the new enhancements, a final stage of quality assurance took place, that led to a series of fixes where required.

The development process of the Real-time group tracking service took relatively more time to complete than the instant message service feature. This feature consists of a few basic additions to the existing Treasure-HIT server side implementation, and a set of new additions to the server, Database and the presentation layers of the authoring platform.

The instant message service was simpler to develop as the mobile application already supported the use of real-time messages, and the Treasure-HIT server side already included a solid web API, which allowed for some simple additions, to enable both the server and the mobile application to allow a new request and response procedure. The mobile applications code is well constructed and easy to understand, which makes it a relatively easy task to add new API calls, functions and data to its different layers. Furthermore, the mobile application includes and supports the use of PhoneGap libraries, which makes it easy to address the different mobile device components and later to compile (code quality validity procedure) and build Android and iOS packages for installation, using Adobe PhoneGap’s online build service.

The third research question focused on the effect the new features on Treasure-HIT users, with emphasis on monitoring efforts of the game-manager and the players’ experience. To evaluate the new features, two main field observations took place with a set of key experienced authoring users and players of different types (students, expert players, QA personnel, software developers and UI designers). After each field observation, qualitative research methodology was used to conduct semi-structured personal interviews, questionnaires and field notes. The first field observation focused on the Real-time group tracking service, which concluded in a set of required UI enhancements and desired map analysis tools. The general feedback from the game-managers was very positive with emphasis on the major need for such a monitoring
tool for this sort of system and the help it provided to better understand the game’s progression, issues and future needed improvements to the game and its content.

The second evaluation included both the Real-time group tracking and the instant message services, which was tried out in a real-time game activity. This field observation included the addition of two different questionnaires that were sent to the game-managers as well as to the players, and was followed by semi-structured interviews. The questionnaires included references to the usability of the features, satisfaction, and a free text option to suggest future additions and enhancements. The instant messaging service, which was examined for the first time on a large scale, received positive feedback in the personal interviews and in the written responses, which indicated satisfaction with the real-time communication effects on the game flow and its impact on the players’ personal feeling towards the game, the learning activity and the content.

In order to summarize the research efforts, which were performed under the different evaluations, I hereby present a summary table that included the reasoning for the evaluation, the participants, methods that were conducted in order to fetch the information, the test parameters and the results, that mostly effected the development of the different feature and their enhancements after the evaluations.

<table>
<thead>
<tr>
<th>Evaluation reason</th>
<th>Participants</th>
<th>Methods</th>
<th>Test Parameters</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Perceptions</td>
<td>▪ 32 first year players, ages 21-30.</td>
<td>▪ Field observation</td>
<td>Usability</td>
<td>Two main requirements: Locations indication, communication between players and game managers</td>
</tr>
<tr>
<td></td>
<td>▪ 3 game-managers</td>
<td>▪ Interviews – semi contracted</td>
<td>Issues</td>
<td></td>
</tr>
<tr>
<td>Real-time group’s tracking map</td>
<td>▪ 4 players</td>
<td>▪ Field observation</td>
<td>Satisfaction</td>
<td>Refinements of the UI, data extension</td>
</tr>
<tr>
<td>tests</td>
<td>▪ 1 game manager</td>
<td>▪ Interviews – semi contracted</td>
<td>Usage</td>
<td>Track poly-line</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>UX</td>
<td>Station/Parent</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Open remarks</td>
<td>graphical indication</td>
</tr>
<tr>
<td>Real-time group’s tracking map</td>
<td>▪ 6 players</td>
<td>▪ Field observation</td>
<td>Satisfaction</td>
<td>Refinements of the UI, indication of received message, limitation of message chars</td>
</tr>
<tr>
<td>tests. Game-manager and players</td>
<td>▪ 2 game-managers</td>
<td>▪ Interviews semi constructed</td>
<td>Usage</td>
<td></td>
</tr>
<tr>
<td>Instant messaging service test</td>
<td></td>
<td>▪ Questionnaires</td>
<td>User experience</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Open remarks</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Holistic</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>perception of</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>the entire</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>solution</td>
<td></td>
</tr>
</tbody>
</table>

This thesis aimed to define new communication options for the Treasure-HIT game game-managers. It follows the development of tools intended to enrich their experience while managing an activity, and to perform later analysis of the structural and pedagogical aspects of the activity, and for the players to experience a more fluid
game experience by reducing their preoccupation with subjects that unrelated to educational activity while maintaining a connection with the game-managers themselves.

The implemented features added abilities to the Treasure-HIT platform that were not possible before, in combining levels between the different users of the system, where analysis can take place both of the construction of the learning activity in the field and of pedagogical matters when analyzing the diverse group actions, including different answer content and number of tries, clue usage and general behavior.

Real-time communication contributed greatly to the efforts of the game managers to monitor and manage the game processes in several ways. It demonstrated the different behavior of game groups under a specific game architecture, helped in producing conclusions on how to make better games and game flow, and to understand the challenges which users face while answering the questions at the different game stations.

Additionally, real-time communication contributed the personal feeling of the players in their efforts to operate within the game by providing them with real-time support during game time on various issues. This helped the players finish the game usefully, relieved the feeling of frustration and isolation when a question was not answered correctly, or when hardware issues required immediate support from the game managers.

Along with the contributions that the two new features add to the platform, it is important to mention that there is much room for improvement. In the case of a large-scale game it can be hard for the game managers to monitor the game flow and the issues raised by the participants during the activity. The reason is that in the current stage of the development of the feature, although the manager is able to monitor each group individually, it is hard to understand the overall perspective of the game. This issue becomes important when bulk problems arise, such as unclear question formulation or GPS issues in certain geo-locations. It will be necessary to find ways to solve these kinds of issues in future developments and adjustments, in order to supply better monitoring solutions for large-scale games. This support should have proper presentation layers of the issues in order to help the game managers analyze and handle the issues.

10. Limitations

The research and development process that took place during the study managed to achieve the goal of creating an environment that contains real-time communication between the teacher and the student, thus providing new information that serves the interests of both types of customers of the system. Despite development achievements, the development process suffered several limitations and difficulties. This section will introduce to two of the main issues that were encountered during the process.

Player’s mobile device and server available resources. To implement a continuous communication with end users there are two methods: opening a socket between the end user and the system server, or long polling (see Appendix A), which sends
continuous requests to the sever till new and relevant information arrives. Both methods are expensive in system resources for the end user’s mobile device as well as for the server. These considerations encouraged the development for both the map and later the instant messaging to base at fixed intervals.

11 Future Work

The current research required two specific developments, real-time group tracking map monitoring and an instant message ability to transfer messages from the game-managers to the players. These features form base for future additions and enhancements that can furthermore enrich the monitoring of the game flow and provide advanced tools for future research efforts. Moreover, adding new features and suggesting improvements to the current implementation over the mobile application can enhance the experience for the players during the game and can add motivation, and support a stronger connection of the student to the context of learning. Here are some new possible features.

- Presentation of all groups on the map – this ability will allow the game-manager to get a general view of all the participating groups in a specific selected game. Every group will be represented by special marker color or image, with a supporting rally track to identify the progress of each group.
- Collective instant message to all groups – adding an ability to send a general message at a specific time to all participating groups can assist in indicating a general issue, provide help, and solve technical problems and physically-related issues. This feature can contribute in saving time for the game-manager in communicating with each group separately. This can also reduce the creation of possible inconsistencies in sending multiple individual messages.
- Summary of a selected group’s locations and actions – Data representation of the group’s track and actions outside the tracking map component. This addition may assist the analysis of group monitoring by providing a way to view the group’s location outside a complex graphical presentation like a map. This feature is mostly relevant when there are multiple markers on the map, a situation that was indicated by the game-managers as a weak point for monitoring and analyzing efforts.
- Fixed graphical gestures for instant messages – sending a graphical gestural form as an image, or a certain emoticon can improve the players’ motivation and indicate to them the level of their performance and achievement during the game. Furthermore, it can assist by providing the game-manager with an unambiguous message, and thus save time in writing a personal message leading to possible confusion on the player’s side.
- Bidirectional communication between the educator and the student – an additional messaging option for the players to communicate with the game-managers can contribute to a faster game, and provide the game-managers with more indications of the events that happen during the game, especially for scenarios that are not being monitored or indicated automatically by the system, and so provide full coverage of all possible events.
- Short intervals for messages updates – the message display on the player’s mobile application is currently subject to fixed actions which are done by the player. For a more continuous game experience and more immediate
connection with the game-manager, the message mechanism should be adjusted to appear when there is a new message available in the system Database. For this, several development options are available, such as sockets and long-polling, all with consideration of the available system resources.

The features suggested here focus on enhancing the monitoring process over groups in a real-time game environment and instant messaging to and from the playing groups. The monitoring will be enriched by being able to see the groups’ progress and their gestures over the map and by the addition of a new channel of communication between the game-manager and the player. Future developments based on the mentioned features can be based on the new and enhanced Treasure-HIT infrastructure, while taking advantage of the code modules that were developed during this research. They were designed to be extended, and they support the principles of SOA, which allows new integrations and expansions for new internal and external services.

Additional research efforts should be made to further explore the effects of the new features on the game-managers and players, to improve the existing features and to open new possibilities for the development and evaluation of new ones.
References


Appendix A: Technologies related to the Treasure-HIT web and mobile applications, development and design patterns

Treasure-HIT Authoring Platform Technologies:

ASP .NET MVC
The Model-View-Controller design pattern decouples different components of a system into autonomic components where each is responsible for its own particular job. This pattern is arranged in three main tiers: The model is responsible for managing and accessing the data, the view is responsible for displaying this data and allowing user interaction, and the controller is responsible for holding the busses login by implementing the required functions for handling user interactions, performing various operations on the back end, communicating with the model when needed and finally supplying the view with the proper data for presentation (Yin, Zheng, Zhang, Zhang, Zhuang, & Ding, 2014).

The ASP .NET framework allows to create MVC projects out of the box with support of all the .NET libraries which allows rapid development, easy implantation, testable code (Microsoft, 2017). The use of ASP (Active server pages) provide an easy way to create an out of the box solution for communicating with the related controller, without the need to create custom AJAX call to the backend services. The ASP MVC uses classic annotations for server side communication and a dedicated implantation of this framework for Razor annotations which allows a server side communication without the dependencies of the .NET framework.

Razor View Template Engine
Razor is a programming syntax that was developed by Microsoft and first implemented under the .NET MVC3 framework. Razor is a view template engine which is used to create HTML content regarding the data that is retrieved from the server and different inputs from the user (Freeman, 2012). Razor syntax is implemented on the client side (HTML pages), which under ASP.NET contains all the regular HTML content, such as JavaScript, CSS reference and HTML tags. The Razor annotations exist on the ASP.NET page itself and contain instructions for the web server where to insert server side-related content (Lydford, 2011).

JavaScript
JavaScript is a client-side programming language which is based on object-oriented principles. JavaScript is embedded into HTML pages, directly or by reference, and is used to enrich the implemented interaction of the user with the page, and make web pages more interactive (Sagar, Duce & Younas, 2009).

AJAX
The AJAX technology (Asynchronous JavaScript and XML), provides a way to implement a REST based web request, which supports both XML and JSON data formats, for different services, receive a response and use the data in a web page (Ying & Miller, 2013). AJAX supports the paradigm of Web 2.0 pages which act as a container, with content being injected to it according to client-based events. These events can vary from click events to text insertion and form submission, and commit an interaction with the server, which in turn, responds with the required data. This unique quality allows for web pages to be refreshed only on specific parts without doing a full-
page refresh and is thus more beneficent for network bandwidth which is saved by refreshing the data only for the required component (Yang, Liao & Liu, 2007).

**Long Polling**
The Long polling method keeps an open HTTP request to the server till new data is available, based on a dynamic interval between requests and the data update frequency on the server (Efremov et al., 2015).

**REST Architecture**
Representational State Transfer (REST), is an architectural style that is used to create client-server communication on different web services (Porres & Rauf, 2011). The REST architecture is based on realizing a set of actions over the unified HTTP interface (Pardon & Pautasso, 2014). Davis (2012) described the following words as the skeleton of the architecture communication protocol that is performed under the HTTP unified interface using those next operations:
- **GET** – Retrieve a resource representation, such as XML, JSON, text or an image
- **HEAD** – This method retrieving data in the same way as GET, but is different in that it does not transform the body of the message itself, for example the getting the message URL.
- **OPTIONS** – Gets information about the communication possibilities with a certain resource, without querying the resource itself
- **DELETE** – used to delete a certain resource that is addressed by the server.
- **PATCH** – Update a resource
- **POST** – The post method is used to address a few required actions: creating a new resource, posting data to the server and posting a message.

**Treasure-HIT Mobile Application Technologies**

**AngularJS**
AngularJS is a JavaScript based framework, maintained by Google, which follows the MVC design pattern in its structure and data flow behavior. The framework implements on the client side, and acts within the DOM of the browser. This creates a special determination of the MVC design pattern, as it facilitates the use of views, which are HTML web pages, and of controller, which is a JavaScript function that handles the data from the view pages and acts as the main component which interacts with the server side Models, using different APIs, that communicate with the system Database (SQL, XML, etc.), and returns the requested data to the client side based controller.

AngularJS is based on a few key concepts that were detailed by Ramos, Valente, Terra, and Santos (2016):
- **Modules** – contain the different components of the application;
- **Templates** – HTML objects which are read by the DOM to create an interface for the user to interact with;
- **Directives** – various text annotations within an HTML page, that directs the DOM to commit specific UI behavior;
- **Expressions** – various code expressions that are used to bind data to the HTML page by curly braces annotations and proper references to declared angular variables
- **Controllers** – used to provide an interface to initialize and update the state of the application;
Digest Cycle – an internal sync process of AngularJS which compares the current value of every variable in the scope elements, and refreshes the relevant components if a new variable value has been updated.

Adobe PhoneGap
PhoneGap is an open source project, maintained and supported by Adobe, which provides a quick solution for developing mobile application for different operating systems. PhoneGap is based on a combination of libraries that are based on JavaScript APIs to communicate with various hardware components on the mobile device. The major advantage that PhoneGap holds for developers is the ability to maintain one code base for the usage of different operating systems, with minimal to no required changes (Amatya, 2013). The code itself is packaged in accordance with the operating system for which the application is intended.
Appendix B: Semi-Structured Interviews Questions

Semi-structured interviews questions – 7.2.1.4 Field observations

Game-managers:
- Under which situations did you find the tracking map feature useful?
- Did you focus on one specific game group or navigated to another using the selection drop-down?
- Does the information presented under both station and information groups markers, helped in the process of analyzing the group progress?
- Have you found the information, under the map markers’ information windows, comprehensible and readable?
- Do the map markers intentions are well understood (from their viewability and position)?
- How often did you use the refresh button? How long do you think that a refresh interval should occur if there won’t be a refresh button exists?
- Did you encounter any technical issues with the tracking map, such as bugs, slowness in data response or missing data?
- Did you find any presented data redundant for the effort of monitoring the game groups in real-time?
- How did you feel, in the notion of monitoring? does the tracking map feature added to the value of monitoring the game (comparing to past experiences)?
- What extra features would you like being implemented for the real-time groups tracking map?

Players:
- What are your general impressions with the mobile application?
- Did you encounter any technical issues with the mobile application? If yes, which?
- Did you encounter any technical issues related to your mobile device? If yes, which?
- What extra features would you like being implemented for the mobile application?
- Do you think that real-time messages, from a game-managers could help with the game general game flow of players?

Semi-structured interviews questions – 7.2.2.4 Field Observation

Game-managers:
- How did you find the instant message feature useful?
- Did you see a change in the groups behavior on the field, after the messages was sent? If so, how much time after the messages was sent, did you notice this behavior and what was the behavior that you see/noticed?
- Did you encounter technical issues with the instant message feature?
- What extra features do you think will be useful for the feature? Do you prefer some property to be changed in any way?
Players:

- Did you receive the instant message from the game-manager?
  - What was the issue which the game manager sent you a message for?
  - Did you find the message helpful to handle your issues correctly?
  - Have you encountered any technical issues with the messages?
  - Do you prefer any other content presentation for the message?
  - Do you think that the message feature contributed to your game-flow and to general feeling of the players while playing?
  - Have you missed the option to write or use any other indication (what kind of indication/flag) that will be sent to the game-manager?

- What extra features do you think will be useful for the instant message feature?
Appendix C: Supervisor and players Questionnaires

Supervisor - HIT Campus exploration activity

Dear Treasure-HIT supervisor,

We would like to present you with a short anonymous survey, to reflect on your personal experience and conclusions, of using the new Groups' location map and Instant messages features that are now available under the Treasure-HIT authoring platform. The survey will be used by us to better understand required enhancements and to prepare future work.

Your help is very much appreciated.

What was your preferred method for monitoring a specific group during game time?

- [ ] Check check point data
- [ ] Monitor the group tracking rally
- [ ] Refresh the map and check groups' markers locations
- [ ] Other...

If you choose other, please explain

Long answer text

Did you use the Instant Message option?*

- [ ] Yes
- [ ] No

Any other comments on the groups' map feature are welcomed ()

Long answer text
Instant Message Feature

Description (optional)

After sending a message to a specific group, did the group react in any physical way that was traceable by the Groups' maps feature? *

☐ Yes

☐ No

Please explain how? *

Long answer text

Except from a free text option, do you like any more additions or maybe a different form of messages options?

Long answer text

Any other comments on the instant message feature are welcomed (relevancy, enhancements...)

Long answer text

Figure 52. Supervisor Evaluation Questionnaire
Player - HIT Campus exploration activity

Dear Treasure-HIT mobile application user,

We would like to present you a short anonymous survey, to reflect on your personal experience and conclusions, of using the new instant messages features that are now available under the Treasure-HIT mobile application. The survey will be used by us, to better understand required enhancements and to prepare future work.

Your help is very much appreciated.

During game time, did you encounter any technical or orientated related issues, that kept you from proceeding with the game?

☐ Yes
☐ No

If you choose yes, please explain what issues did you encounter

Long answer text

If you encountered any technical or orientated related issues, did you get a message from the supervisor relating to the matter?

☐ Got a message from the supervisor
☐ Got a message from the supervisor but it wasn’t related to the encountered issue
☐ Got a message from the application
☐ Didn’t get any message from any type

What was the message content? *

Long answer text

Any other comments on the instant message feature are welcomed (relevancy, enhancements...)

Long answer text

Figure 53. Player Evaluation Questionnaire