

Challengelt: Building the 21st century classroom economically

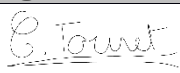

CHRISTOPHE TOURET

EMBRY-RIDDLE AERONAUTICAL UNIVERSITY

GRADUATE RESEARCH PROJECT

04/04/15

VERSION 1.5

	Name	Signature	Date
Student	Mr. Christophe Touret		4/7/15
GRP Advisor	Dr. Shafagh Jafer		4/7/15
MSE Coordinator / Department Chair	Dr. Remzi Seker		

Revision History

Date	Version	Description
05/05/14	0.0	Document creation. First draft of Abstract, Introduction, State of the art.
05/06/14	0.1	Use cases.
05/07/14	0.2	Functional Analysis.
30/07/14	0.3	Design model.
01/13/15	0.4	Review of the Abstract and Introduction
01/25/15	0.5	Project Requirements.
02/02/15	0.6	Design.
03/21/15	0.7	Review of project requirements. Backend API, prototype section started.
03/29/15	1	Completion of document (first draft).
04/02/15	1.1	Review of the document. Acknowledgments.
04/04/15	1.2	Review of Acknowledgment, Abstract, Introduction, State of the Art, Ergonomy, Future Work and Discussion and conclusion.
04/05/15	1.3	Review of Introduction, State of the art, Project Requirements, Design and Prototype.
04/06/15	1.4	Review of the entire report.
04/07/15	1.5	Signature of GRP Advisor and student.

ACKNOWLEDGEMENTS

I would like to thank my GRP advisor Dr. Jafer that helped me to define project subject, gave me good directions during the project, and encouraged me to pursue with this project into great opportunities like the IBM/IEEE Smarter Planet Challenge 2014 and the IEEE SoutheastCon 2015.

Thanks to my friend Eric Truong that made the ChallengeIt mock-ups look great using Photoshop.

Thanks to my friend Marc-Antoine Latil for participating in the elaboration of the project with me in the early phase and without who I would not have had the idea of the app.

Thanks to my friend Ryan Gauthier for participating with me at the Smarter Planet Challenge 2014. Some contents of this report were readapted from the Smarter Planet Challenge report in which he wrote a lot with me. Thanks for the rest of the Smarter Planet Challenge team that came afterward, especially to Janani Thiagarajan that did most of the work for us to have video¹.

Thanks to Dr. Bower that advised me personally to make this project bigger on the way of entrepreneurship. He also helped with Dr. Jafer to make the team for the Smarter Planet Challenge 2014.

Thanks to my friends from LYFEBOAT, Chad Gardner, Phat Le and Michael Reha with who I was interning during my last semester, and who allowed me to work in parallel on my thesis with a schedule that suited me the best. Thanks also to Michael Reha, Chad Gardner and Dr. Jafer who helped me review this report.

¹ The video is available on Youtube at the following link: <https://www.youtube.com/watch?v=FnoIvo0firo>

ABSTRACT

This graduate research project aims to tackle the lack of student involvement and motivation in our nowadays classroom. What if the boring classroom experience can become an immersive, effective and engaging game for both the student and instructor? What if a student could be motivated to accomplish more than just the learning requirements of a class? What if the way students interact in a class matches the way they interact with their friends? To make these things real, we created **Challengelt**: a mobile application that allows students to compete with one another on challenges in their classrooms in a fun and interactive manner supported by mobile learning. The classroom can now be supplemented with a targeted “game”, where students can earn points by completing challenges, and having their instructors review those answers and award points. Furthermore, students are free to use any medium or combination of media with which they prefer to respond to these challenges (e.g. text, PDFs, photos, videos, etc.). This will make them able to always participate in the class (both at school and at home) in a way that makes them most comfortable. If the instructor so chooses, real-world problems could also be introduced into our solution, thereby allowing student efforts to be channeled into real-world solutions.

Challengelt represents an innovative solution because it represents a fusion of the edutainment game genre and the interactive classroom tool. It is economical to implement, easy to use, and provides convenient access to learning activities wherever the student is at any time. **Challengelt** also represents a departure from previous attempts to implement game-based digital classroom. The Classroom Response System from Carnegie Mellon University employed a proprietary response system designed for use inside the classroom [1] and the WE Learn global initiative by Qualcomm Incorporated tried to deploy large amounts of highly customized and specialized 3G smartphones to connect students and instructors wherever they might be [2]. This latter solution is very recent and successful, indicating that the idea of using smartphones for education has merit and potential worth exploring. We are hoping to achieve similar results with **Challengelt** by incorporating an additional fun-factor, but at a much lower cost to students, parents, and schools. This research project introduces the interactive game **Challengelt**, and covers the analysis and design, and development details from a software engineering aspect.

Contents

Acknowledgements	3
Abstract.....	4
Introduction	8
State of the art	10
WE Learn by Qualcomm.....	10
Serious games in the classroom.....	10
Interactive classroom tools.....	11
Apps in the classroom:.....	11
Conclusion on the literature:.....	11
Project requirement.....	12
Introduction.....	12
Purpose	12
Scope	13
Definitions, acronyms, and abbreviations	13
Stakeholders.....	16
Overall description.....	16
Product perspective	16
User characteristics	19
Constraints.....	19
Specific requirements	20
External interfaces.....	20
Functions	20
Performance requirements	22
Logical database requirements.....	23
Design constraints.....	23
Ergonomy of ChallengeIt.....	24
Story Boards	24
Message Creation.....	25
Game.....	27
Create Game.....	28
Review challenge response	29
Edit profile and settings.....	29
Design model	30
Google Cloud Platform	30
Datastore.....	31
Cloud Endpoint.....	34
Prototype presentation.....	36
Connecting to the app	36
Main Activity.....	37
Master	38

Player	39
Friend.....	40
Create Game	40
Future work.....	42
Discussion and conclusion.....	43
Appendix.....	44
Sequence Diagrams.....	44
References.....	47

Table of figures

Figure 1 Conceptual Diagram	15
Figure 2 Uses cases diagram	18
Figure 3 External Interfaces of ChallengeIt	20
Figure 4 Logical Database Requirement ChallengeIt	23
Figure 5 Overview of the app views structure	24
Figure 6 Create Message Modes	26
Figure 7 Games and challenges lists	27
Figure 8 Interaction between functionalities, views and objects of the games	28
Figure 9 Create Game	28
Figure 10 Manage friends	29
Figure 11 Edit profile and setting	29
Figure 12 Review Response	29
Figure 13 ChallengeIt architecture overview	31
Figure 14 Pricing Datastore [18]	32
Figure 15 Datastore structure	33
Figure 16 Google cloud endpoint architecture [19]	34
Figure 17 UserLocal API	35
Figure 18 Game API	35
Figure 19 Sign up with Google Account	36
Figure 20 Connection to the app	36
Figure 21 Main activity fragments	37
Figure 22 Master Game Activity	38
Figure 23 List Responses Activity	38
Figure 24 Response activity	38
Figure 25 Create Challenge Activity	39
Figure 26 Player Game Activity	39
Figure 27 Challenge Activity	39
Figure 28 Create Response Activity	40
Figure 29 Friend fragment	40
Figure 30 New game item	40
Figure 31 Create New Game Activity	41
Figure 32 Add a friend sequence diagram	44
Figure 33 Add a game sequence diagram	45
Figure 34 Add a player sequence diagram	45
Figure 35 Create message sequence diagram	46

INTRODUCTION

Lack of student engagement and motivation in modern-day classrooms is an important problem in the United States and one way to look at it is by understanding its economic impact. Dropout is the confirmation of the real lack of student involvement, and a study shows that in Wisconsin, the dropout rate of high school students is about 10 percent each year in the last decade [3]. Already by analyzing the school dropout rate, we can understand that the economic burn is gigantesco: economics professors Henry M. Levin and Cecilia E. Rouse concluded in their study that, by reducing the number of dropouts by just half, there would be close to 90 billion dollars saved for the government per year in the United States [3].

One reason this could be happening might be the increased prevalence of technology in our day-to-day lives and how it has made us unable to focus on any particular task for long periods. Why not use these technologies to make them more focused instead? Technology has also introduced fundamental new forms of communication: forms that are ingrained in young adults from a very young age and represent their most comfortable manner of interaction. Essentially, adapting school to address these issues would require a fundamental paradigm shift in the very way we prepare and conduct education globally.

Indeed, up-coming generations of students are real digital natives, technology is part of their everyday life, and they are immersed in interactive environments using social networks or instant messaging applications. Current study suggests that Net Gen students are more likely to engage digital interaction than to interact with other students or the instructor when in a face-to-face learning environment. However, despite the waves of new technologies, the educational system remains mostly intact and does not provide the expected technologies in the classroom. Besides, nowadays, it is simple to enhance education with digital interaction for Net Gen's students because most of them own suitable devices like smartphones or tablets [4].

In parallel, current study shows that digital gaming has largely unrealized pedagogical potential. Indeed, game-based-learning forces students to take an active approach to learn with rapid feedback and clear consequences leading to higher engagement and improved learning [5]. These games, which are designed for serious purposes like education, are called serious games [6].

Serious games seem to be a promising approach for these purposes, and in fact, the serious game market is blooming with an impressive growth. According to a study from the Audiovisual and Telecommunication Institute of Europe (IDATE), world Serious Game market, which was worth 2 billion dollars in 2010, will be worth 14 billion dollars in 2015 [7].

Challengelt looks to utilize technology that most students already have, and to use it to create an interactive and fun environment to further studies both within and outside the classroom. There already have been a number of commercial successes in fitness, task completion, and healthy living, like this article from Smith and Jones attests: “Hospitals and physician practices are starting to use social media to incentivize patients to lead healthier lifestyles through integrated online and mobile games as a means of encouraging competition between individuals to foment personal motivation” [8]. More specifically, in the field of education, the WE Learn global initiative by Qualcomm Incorporated tried to deploy large amounts of highly customized and specialized 3G smartphones to connect students and instructors wherever they might be [4]. This latter solution is relatively recent and was relatively successful, indicating that the idea of using smartphones for education has merit and potential worth exploring. **Challengelt** is aiming to do the same for education, and it will be fast and inexpensive to set up.

The main contribution of this research project is to introduce the project **Challengelt**, a step forward in building the 21th Century classroom as the project WE Learn global initiative by Qualcomm Incorporated intends to do, but in a more fun and cost effective way. This research is seeking to expose our vision of tomorrow’s classroom through **Challengelt**.

Seeking that goal, this document introduced firstly the current state of the art for building the 21st century classroom. Secondly, it presents formal requirements and suggests an ergonomomy of **Challengelt**. Then, it presents a technical design for **Challengelt** with a description of the technology used to meet the requirements and to be cost effective. It is also presenting a prototype proof-of-concept of these suggested technologies and a framework for the whole app to be actually implemented. Finally, it exposes the future work to complete the project, and open a general discussion about the project.

STATE OF THE ART

The largest technology-based companies like Microsoft, Apple or Qualcomm are already investing in how to build the 21st century classroom [9, 10, 2]. Additionally, a large amount of educational serious game and learning mobile apps are trying to improve the classroom experience. This section is aiming to draw a non-exhaustive but relevant picture of how nowadays classrooms can use digital technologies.

WE LEARN BY QUALCOMM

The WE Learn mobile education is aiming to accomplish similar objectives than the ones of **Challengelt** by providing smartphones working on 3G with a variety of educational software to young students. The project is seeking to transform learning from a traditional and instructor-centric model to a student-centric, inquiry-oriented and collaborative model. By enabling 24/7 access to resources in and out of the classroom, the project allows students to acquire and practice 21st century competencies and knowledge [2].

The WE Learn pilot project was a success and demonstrated the following improvement in education [2]:

- The study coming along with the project shows that students have improved self-directed and collaborative learning skills. More specifically, they developed better ability to thrive in a fast-changing and highly connected world.
- Instructors have adapted their teaching methods to incorporate technology and reflect the positive results and responses from students.
- Parents became more open to use and support technologies for learning like tools that make the student able to record text, drawings, audio and video.

Challengelt is aiming to reach the same results but in a fun, cost-effective and much more open way. Qualcomm provided students with smartphones using their hardware technology and software coming specifically from Microsoft, e.g. MyDesk. They are providing a proprietary solution that aligns with their interests: advertising and promoting their technologies.

Challengelt wants to offer an open solution that allows the students to use their own smartphones. Furthermore, **Challengelt** adds a fun aspect to the classroom experience being game-based, which may increase the student acceptance of the system.

SERIOUS GAMES IN THE CLASSROOM

This subsection is considering as examples two edutainment games integrated in classroom: one made for business major students from universities named “Innov8”, and one made for young students in middle school named “Kids and Cookies”.

- Innov8 by IBM is a serious game that simulates a business environment to illustrate how Business Process Management (BPM) could be used in the professional world [11].

- Kids and Cookies is an interactive game designed for elementary students. The game is an interactive way of introducing number concepts. Users simulate sharing cookies while investigating the concepts of rational numbers and division [12].

As it the case for these two examples, edutainment games address specific subjects and they incorporate the specific knowledge that the user should learn on the subject. The serious games, being as complex as the subject they are addressing, could easily become quite expensive to develop, especially for high level and complex . Furthermore, in the two examples previously presented, the instructor has no control upon the interactions.

INTERACTIVE CLASSROOM TOOLS

Microsoft conceived an interactive classroom, which is a PowerPoint, and OneNote add-in that allows the instructor to insert customized classroom polls and notes into his presentation in real-time [13].

However, this proprietary tool is not containing any gaming or entertaining aspect, and is not suitable for student's mobile devices. Other interactive tools for classroom like interactive board are expensive, and remain purely academic (they do not contain an entertaining aspect).

APPS IN THE CLASSROOM:

The success of mobile applications in education can be proved by their numbers, for example, App Store is recording 65 000 apps for education. Mobile apps in the classroom is an up-growing market that shows how much education is open for integrating new apps in the classroom. Apps in the classroom are usually either simple serious games specific to a subject, or utility apps that are used as a learning tool. As we have seen previously, serious games have specific knowledge incorporated into the app, and allows the student to study or train for a specific subject. Utility apps can help the student during class like Moment Diary, which is private diary that assists the student to collect learning contents [14]. These utility apps do not contain any game aspect and are often made for a specific purpose. In addition, most of these apps do not offer an interactive system where the instructor has control over the class; the apps aim to replace the instructor instead of supporting him.

CONCLUSION ON THE LITERATURE:

By analyzing the different attempts for integrating digital technologies in the classroom that were reviewed, it seems that the current entertaining approaches, i.e. the serious games, are trying to replace the instructor for a specific training, or learning, more than to provide an interactive environment under the control of the instructor. On the other hand, interactive tools and utility apps remain utilitarian and academic, and do not include the advantages of the game approach. Furthermore, for both serious games and apps in the classroom, it seems that the more complex the learning subject, the more complex and expensive the solution is to be developed. No solutions offer a game-oriented, interactive, instructor-monitored, open and cost-effective system, and that is why *ChallengIt* was created.

PROJECT REQUIREMENT

The underlying section is the System's Requirement Specification (SRS) of the **Challengelt** project based on the IEEE 830 standard.

INTRODUCTION

PURPOSE

Challengelt aims at responding to the following main concerns:

- ➔ Enhancing student's involvement and motivation by using a game-based environment.
- ➔ Providing a learning style suited to Next Gen's students by making them use their everyday mobile device in a similar way of what they are used to.
- ➔ Allowing communication with a variety of format including text, files, photos, audio and videos.
- ➔ Keeping the class interactions under the instructor's monitoring.
- ➔ Being a platform suitable for all kinds of classroom context.
- ➔ Being simple, affordable and easy to set up.
- ➔ Motivating students to continue learning outside of the classroom by providing challenges that expand upon what is learned in the classroom, encouraging ideal learning [15].
- ➔ The ability to receive feedback from the instructor or the students' classmates.
- ➔ Using a RESTful cloud-based system for the backend to have scalability, low-cost and efficient expandability to different client type (web app, iOS app).
- ➔ Allowing real life issues to become students' challenges.

The underlying project, namely **Challengelt**, proposes an innovative approach based on challenges for creating an interactive game-based application to integrate in the classroom.

Challengelt is harnessing the power of challenges with an interactive mobile application. As current study explained, educators should use challenges in a serious gaming context to improve learning [15]. However, unlike most serious games, **Challengelt** will not contain any incorporated knowledge. Our contribution resides in establishing an instructor-monitored interactive platform that emanates a challenging dynamic for the student.

For ensuring its success and acceptance among students, the application will incorporate a similar ergonomic navigation that the popular apps such as SnapChat and Facebook Messenger implement. The app also allows for a vast variety of input and multimedia to be used to create and complete challenges.

With **Challengelt**, the class changes into a game. Users will play games, e.g. be in classrooms, and each game is going to be monitored by a game master, e.g. the instructor. In each game, the master broadcasts challenges to the players of the games, and by winning the challenges the players earn points. Each challenge is timed, and the winner is the player who has the most points at the end of each challenge. Within a game, players are constantly ranked based on the number of points they have. Allowing the ability for classmates to compete with one another increases student involvement.

In parallel, it is also a tool for the student to collect and share their work and feedback in one environment with an adapted format. The messages used for communicating inside that app will be created using an innovative and adapted fashion: they should be created using both a media that a smartphone can record, e.g. picture, audio and video, and what is allowed by email, e.g. the use of text and attached files.

Separately, in high-level classes, external actors might use **Challengelt** for suggesting their own challenges to the students. In that way, problems of the real world might become the challenges of the students.

Concretely, **Challengelt** will be a game-based and specifically organized communication tool that students can download on their Android device. **Challengelt** should be implemented on Android because Google Play is today the most widely used mobile platform and it is the most accessible considering the lower cost of most Android devices.

SCOPE

The scope of this project includes a software application on Android called **Challengelt** app and the backend on Google App Engine called **Challengelt** backend. The functionalities and the qualities to be met of the project are described in the Section 3. The benefit of the project is to improve student involvement in classes.

The project does not include:

- Any research to prove the efficiency of the method.
- The platform in which the app can be download (we will use Google Play).
- The smartphone or the OS (Android) of the smartphone in which the App should be installed.
- Any Android library used or services provided by Google App Engine to support the backend of the app.

DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

Term	Definition/Description
OS	Operating System
GAE	Google App Engine
Backend	Abbreviation for Challengelt backend
app	Abbreviation/ synonym for Challengelt app
User	A person that is using the application. Each user is uniquely identified with a username . The username is the email address of the google account he signed in to the app with.
Master or game master	A user with specific rights that allow him to monitor the game. A master creates and set up the game, invite players, approve challenges, rate the challenge response and edit the game he is mastering if needed. A master would typically be either a professor,

	a tutor or a chief of project.
Player	A user that is playing on a specific game. A user needs to be invited by a game master to join a game, and can be in multiple games simultaneously. Only friends can be invited to be play at a game. When invited to a game the user become automatically a player of the game. However, the player is able to leave a game whenever they want.
Message	A message can be a media, text, attached files or a combination of them.
Media	A media is a either a photo, a video or an audio record.
Profile	A profile is the set of attributes characterizing and identifying a user.
Game	<p>Every user can create a game and become the master of it. A game should be created with at least a title, one a player and a mode. A game can be created with two different modes:</p> <ul style="list-style-type: none"> • Public mode, in which, for each challenges of the game, players' submissions are accessible by the other players of the game. • Private mode, in which, for each challenges of the game, all players' submissions are not accessible by the other players of the game. <p>A game can also be created with a description and several users. During the game, the users will be able to create and respond to challenges under the monitoring of the master. In each game, players are ranked considering the number of points that earned by completing challenges: the more they have points, the better they are ranked.</p>
Game settings	Each game can have settings: a title, a description and the mode of the game (public or private). In the future, different modes could be added to the game settings.
Rating	An assessment of a player submission made by the master. The master "rates" the submission with a number of stars that can be from zero to five.

The following conceptual domain model describes the concepts used in **Challengelt**:

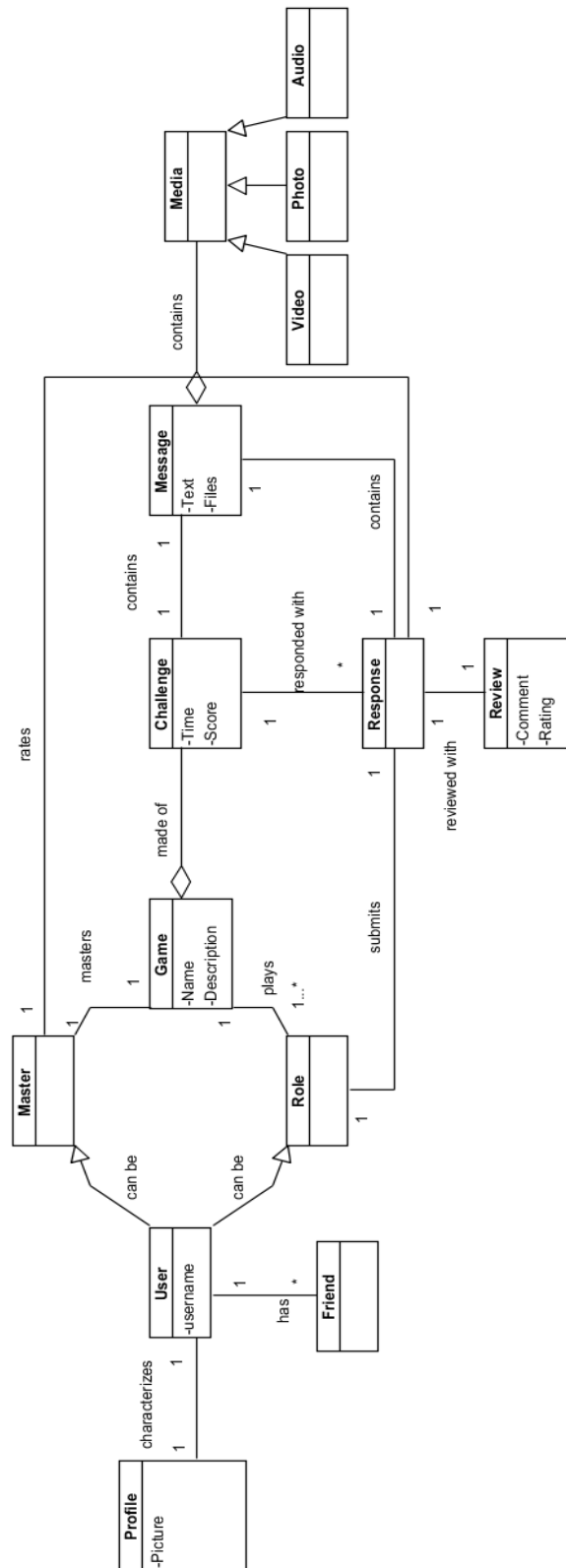


Figure 1 Conceptual Diagram

STAKEHOLDERS

User: A person using the app. When playing at a game, can be either a Master or a Player.

Master: A user monitoring the game, e.g. the instructor of the class.

Player: A user playing at a game, e.g. a student of the class.

School: The institution in charge of providing Android Device to students that do not have one.

GAE: The service provided for the backend of the app.

Android: The OS on which the app will be.

Google Play Services: The service that enables authentication of a user into the app using one of the Google Account on his phone.

Google Play Store: The platform from which the app could be downloaded.

OVERALL DESCRIPTION

PRODUCT PERSPECTIVE

The **Challengelt** project is supported by the Google Cloud platform for the backend and Android for the frontend. Students and instructors will use the app to, respectively, play and monitor **Challengelt** games.

User Interfaces

The user interfaces are defined in the section “Ergonomy” of this report.

Software interfaces

Software	Minimum version supported	Target version
Android	API 16	API 21
Google Play Services	v6.5.87	v6.5.87
GAE	v1.9.17	v1.9.17

Communication interface

Wi-Fi, 3G, 4G can be used for communication between the clients and the backend.

Product functions

This section is exposing the main top-level requirements of **Challengelt**. The Figure 2 presents the use cases diagram of **Challengelt**.

The system shall:

- Allow a user to manage their friends, e.g. invite and accept invitation from users to be a friend, and delete friend.
- Allow a user to create a game with specific game settings (she will then become the master of the game) and invite users to be player of the game.
- Allow a player to suggest a challenge to the master of the game.

- Allow a master to edit the game settings, add and delete players of the game and create, approve, and edit challenges.
- Allow the master to weight points and set a deadline to submit responses.
- Allow a user to create a message to create a challenge, or respond to a challenge, using a media like video, photo or audio, files or text or any combination.
- Allow the master to provide a score and a comment to the feedback for each submitted response.
- Record each submission of the users.
- Allow the user to access the information of the games and challenges in which he is involved.
- Determine the rank of the players of a game and the winners of the completed challenges.

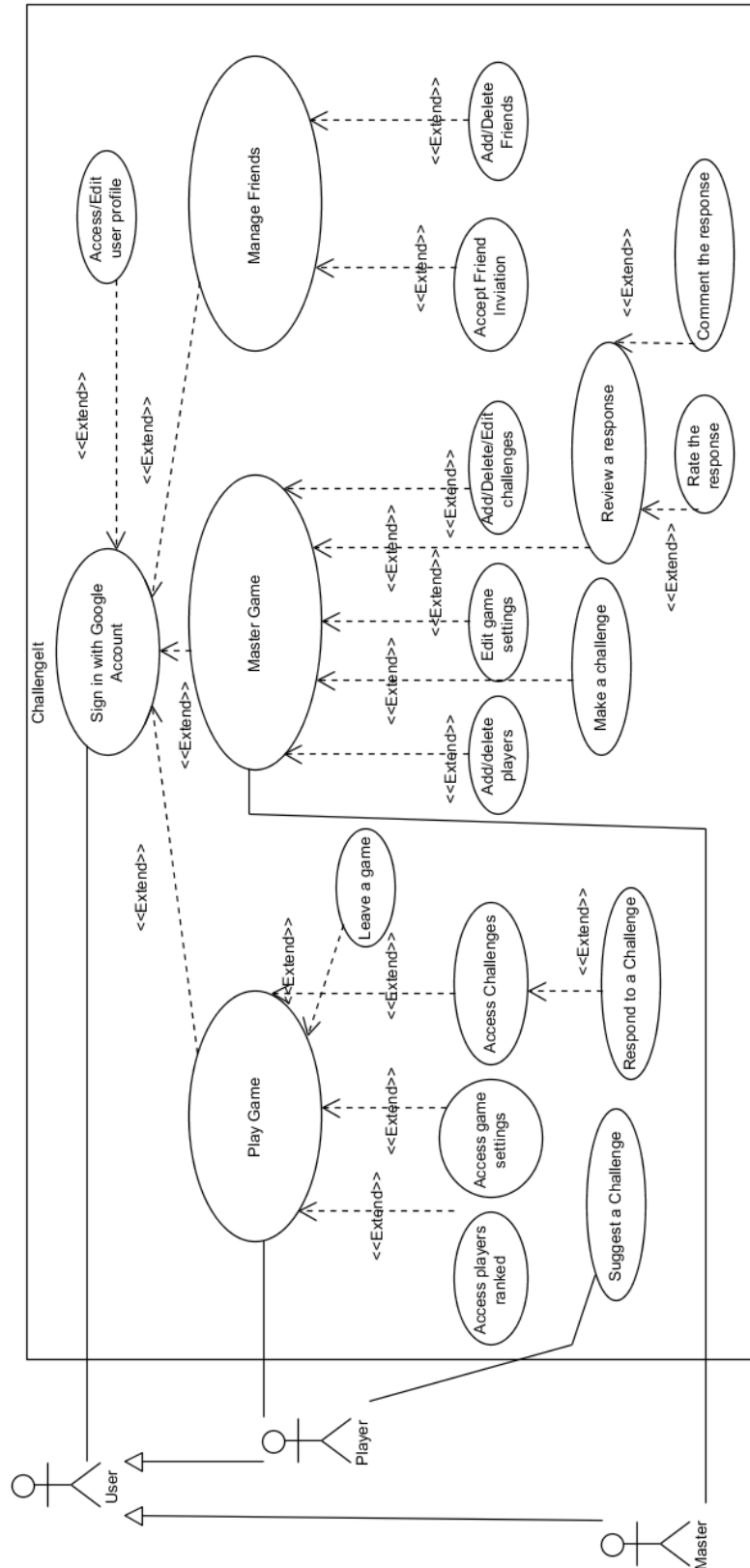


Figure 2 Uses cases diagram

USER CHARACTERISTICS

User	Description
Student	Any student from elementary school to university that possess and is able to use a proper Android device for the app.
Instructor	Any instructor from the educational system that has access to an Android device.

CONSTRAINTS

Hardware constraints

	RAM	CPU
Minimum capacity of the Android device	1 Gb	1.2 Ghz
Best characteristics of the Android device for usage	2 Gb	2.5 Ghz

The phone shall have a camera integrated.

Communication constraint

The user shall have a network that allow download of important files as video. The files shall not be bigger than 5 GB. The network shall be able to download that amount of data in less than 90 seconds.

SPECIFIC REQUIREMENTS

EXTERNAL INTERFACES

As the Figure 3 shows, the system has an interface with GAE, Android and both Master and Player users.

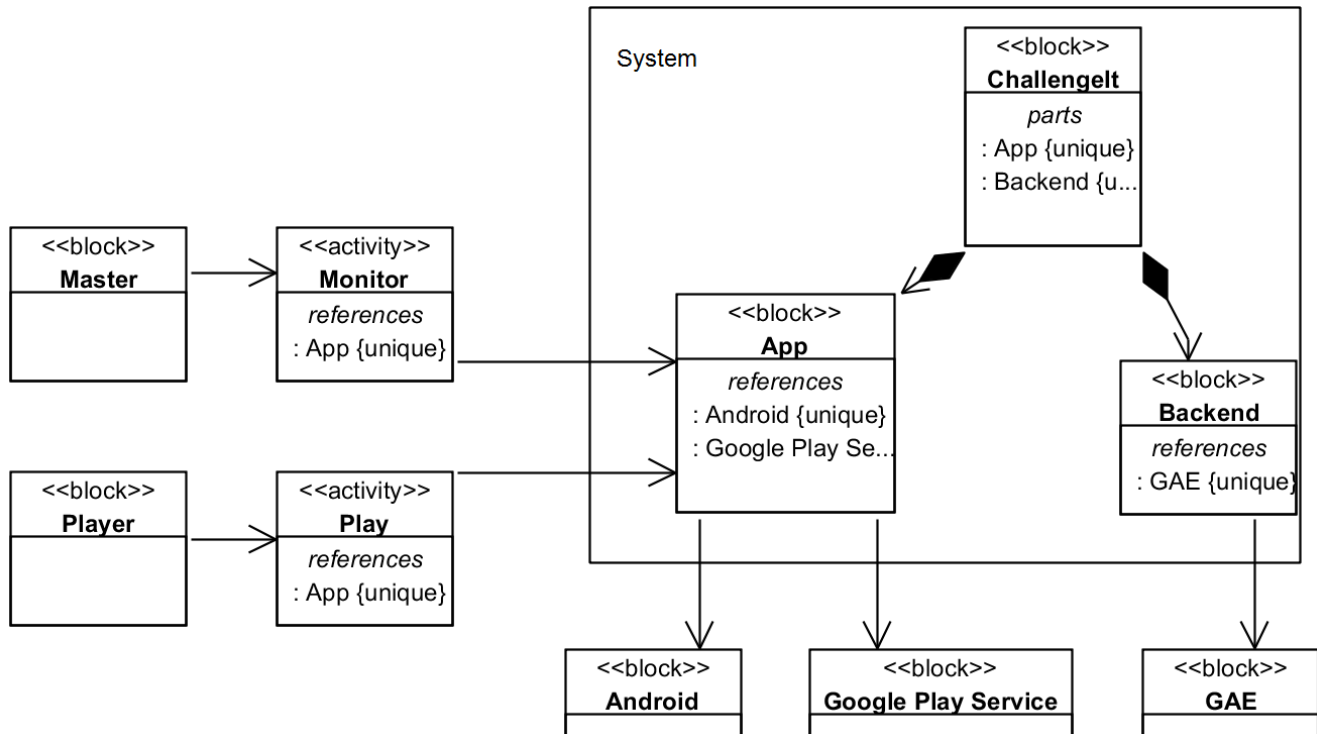


Figure 3 External Interfaces of ChallengeIt

FUNCTIONS

Preconditions: The user has already downloaded and installed the app into a compatible Android phone. The user has already a Google Account and Google Play Services set up in the Android phone.

1. The system shall authenticate a user with an OAuth protocol.
 - 1.1. The user shall sign in into the app using a Google Account to be able to use the app.
 - 1.2. The user shall be able to choose between the Google Account already registered in his phone when signing in.
 - 1.3. The user shall be able to disconnect from all services.
2. The system shall allow a user to manage his friends.
 - 2.1. A user shall be able to invite a user to be his friend on the system using his username.
 - 2.2. The system shall notify a user when he received a new friend invitation.
 - 2.3. A user shall be able to accept or refuse a friend request on the system.
 - 2.4. The system shall add into the user friend list all users that has accepted his friend request.

- 2.5. The system shall add into the user friend list all users to whom he has accepted the friend request.
- 2.6. A user shall be able to access his friend list.
- 2.7. A user shall be able to access the user profile of his friends.
- 2.8. A user shall be able to delete friends from his friend list.
3. The system shall allow a user to create a game.
 - 3.1. The user shall at least have one friend to create a game.
 - 3.2. The user shall choose at least a game title and a friend to invite into the game.
 - 3.2.1. The system shall automatically add the invited user to the game as a player.
 - 3.3. The user shall be able to add a description to game.
 - 3.4. The user shall be able to add friends of him into the game.
 - 3.5. The user creating the game shall automatically become the master of the game.
 - 3.6. The user shall be able to select either the public or the private mode.
4. A player shall be able to play a game he has been invited to.
 - 4.1. The player shall be able to access to the game settings of the game.
 - 4.2. The player shall be able to access to the rank of players of the game.
 - 4.3. The player shall be able to access to every accumulated score of the players for the game.
 - 4.4. The player shall be able to access to the challenges of the game.
 - 4.4.1. The player shall be able to respond to challenge using a message if it is before the deadline of the challenge.
 - 4.4.2. The player shall be able to access to the content of each challenges.
 - 4.5. The player shall be able to suggest a challenge for the game.
5. A master shall be able to monitor the games he has created.
 - 5.1. The master shall be able to edit a game
 - 5.1.1. The master shall be able to edit the game's settings.
 - 5.1.2. The master shall be able to add player(s) into the game.
 - 5.1.3. The master shall be able to delete player(s) from the game.
 - 5.1.4. The master shall be able to create a challenge.
 - 5.1.4.1. The master shall add a title to the challenge.
 - 5.1.4.2. The master shall be able to add a description to the challenge.
 - 5.1.4.3. The master shall be able to add a deadline to the challenge.
 - 5.1.4.4. The master shall be able to add points to the challenge.
 - 5.2. The master shall be able to review the challenge' responses.
 - 5.2.1. The master shall be able to rate the responses from zero to five stars.
 - 5.2.1.1. The player will receive a number of points of the challenge proportional to the rating. The points to be received by the player shall be calculated using this formula :

$$\text{Points for player} = \frac{\text{rating}}{5} * \text{number of points of the challenge}$$

5.2.2. The master shall be able to comment on the responses.

6. The system shall allow a user to create messages to create or respond to challenges.
 - 6.1. The user shall be able to create a message first and then decide to create or respond to a challenge.
 - 6.2. The user shall be able to use the message to create a challenge.
 - 6.2.1. If the user is a player, the challenge created shall be sent to the master for approval
 - 6.2.2. The master shall be able to approve the challenge
 - 6.2.2.1. The system shall request the master to weight a challenge with points once approved.
 - 6.2.2.2. The system shall allow the master to affect a deadline to the challenge.
 - 6.2.3. If the master approved the challenge, the system shall broadcast the challenge to the students.
 - 6.3. A player shall be able to use the message to respond to a challenge.
 - 6.3.1. The system shall request the master to provide score feedback to each submitted response.
7. The system shall determine the winner of a challenge after the deadline is passed.
 - 7.1. If many players have the same rate for the challenge, the best-ranked player is the one winning the game.
 - 7.2. The system shall expose the name of the winner for the challenge.
 - 7.3. The players of the game shall be able to access to the winning response.
8. The frontend shall be made in accordance with the ergonomics of *ChallengeIt* section of this document.

Note: 'R' followed by the number of the requirement will refer the above requirements.

PERFORMANCE REQUIREMENTS

1. A game shall allow up to 50 players.
2. The system shall support message weighting up to 1 Gb.
3. The user shall have the updated in the app every time he is connecting to the app.

Note: 'P' followed by the number of the requirement will refer the above requirements.

LOGICAL DATABASE REQUIREMENTS

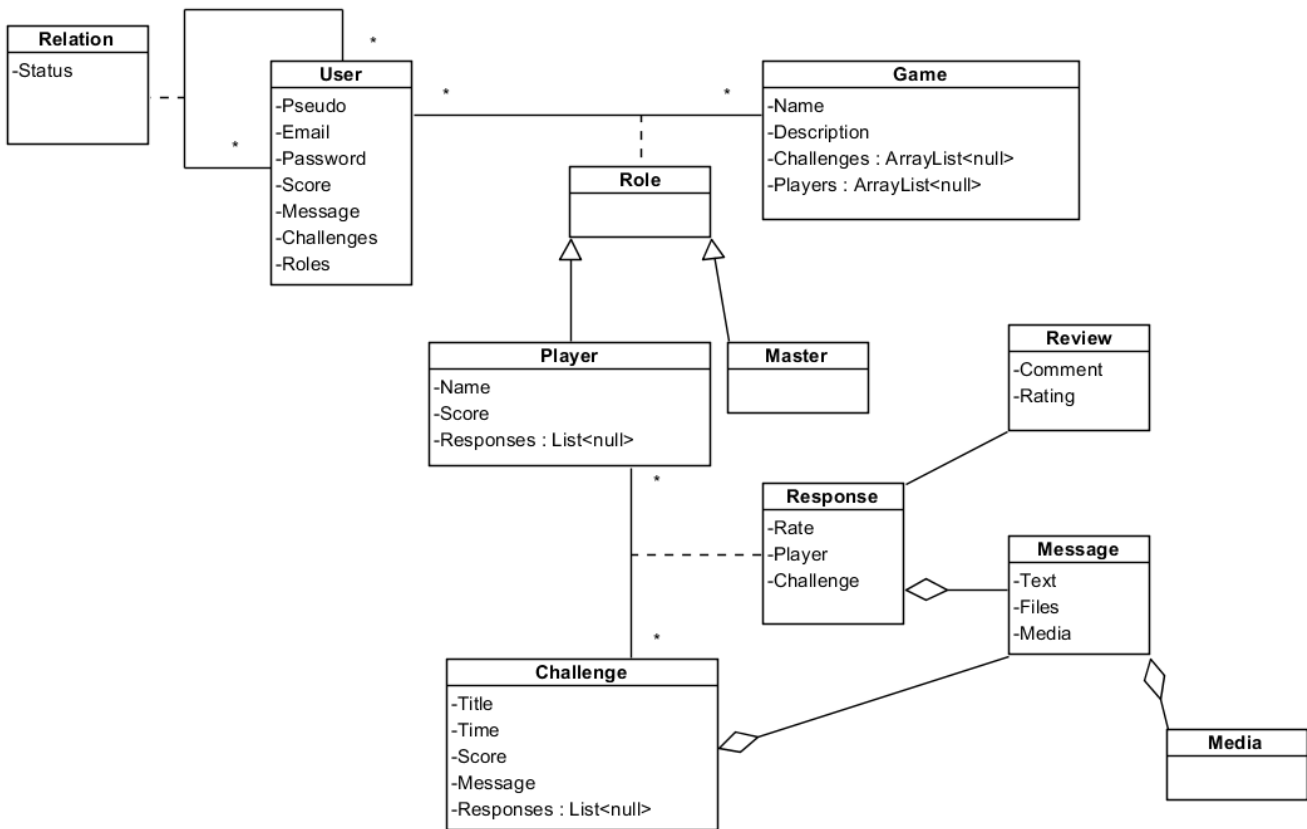


Figure 4 Logical Database Requirement ChallengeIt

DESIGN CONSTRAINTS

In order to be economical and maintainable, the design shall be respect the following constraints:

- The backend should be done using GAE services.
- The persistence to the Datastore should be done using the Objectify framework and cloud Endpoint technologies.
- The app should be made for an Android environment.
- The app should follow the best practices guidelines from Android.
- The authentication should be made using Google Play Services.

ERGONOMY OF CHALLENGEIT

STORY BOARDS

OVERALL PICTURE

In order to use the app, users have to sign into the application with a Google account. When first using the app, users have to choose which Google account they want to use to sign in with beyond the accounts saved into their phone. The email address associated with the Google account will become the username of the **ChallengeIt** account automatically created on the first connection. Because we are using the email address associated with a Google account, the user and owner of the Google account should give us the permission to access to his Google account email at the first connection. Then, the user can access the app and navigate through its views as described in the diagram below.

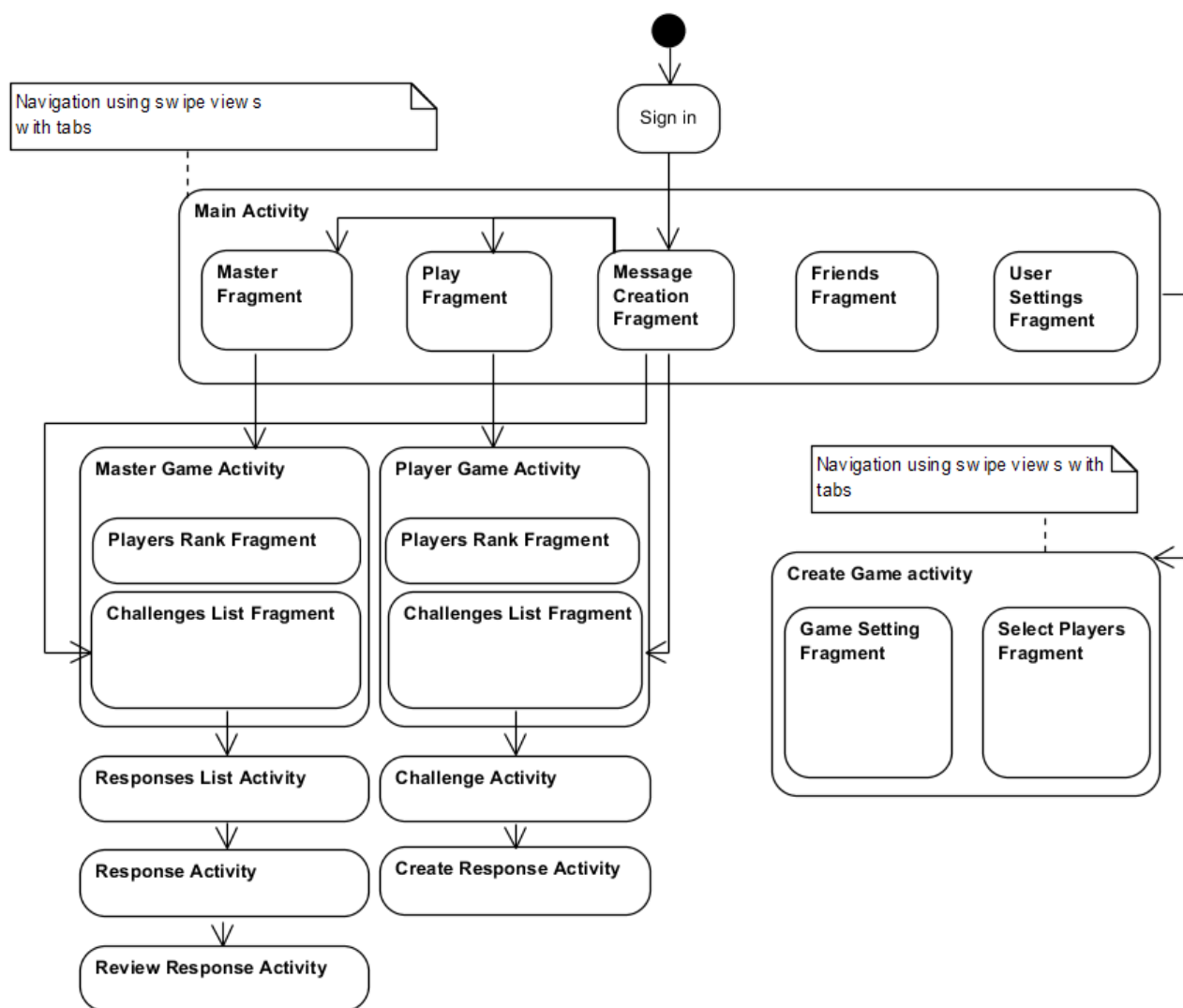


Figure 5 Overview of the app views structure

MAIN PANEL

An authenticated user will first access five different fragments from the main activity:

- one frame will show a list of all the ChallengeIt games that the user is mastering,
- one frame will show a list of all the ChallengeIt games in which the user is playing,
- one frame will allow the user to create a message by either taking a picture/video, sharing a file or writing text,
- one frame will allow the user to send friend invitations to other users and will allow the user to accept or refuse friend invitations from other users,
- one frame will allow the user to access and edit his profile information.

The model used to navigate between these frames shall be a swipe view with tabs. Each frame would correspond to one tab.

MESSAGE CREATION

In the same spirit of popular apps like Instagram or Snapchat, ChallengeIt was conceived to be a message-oriented app. The user shall be able to create the message first and then decide what he wants to do with it, e.g. create a challenge or respond to a challenge. The user should also be able to do the opposite, e.g. first go to the game to create a challenge, or first go to a challenge to create a response.

One of the main innovations of ChallengeIt will be its new way of creating messages. A message should allow the user to use the variety of media offered by the mobile devices combined with the usual functionalities of the email system. Thus, the message creation is composed of two parts:

- A media fragment, in which users can take a picture, a video or a voice recording.
- An email fragment, in which users can write text, and attach files.

When opening the app, the user starts with the message creation fragment on the full camera mode (a). From this mode, the user can start recording a media or swipe down the dashed T for access to the email part. If he swipes the dashed T to the bottom of the screen, he will be on the full email mode (c). He can go back to the full camera mode by swiping the dashed T to the top of the screen. Once he takes a media (photo, video or audio), the user can enter on the dual mode (d) by swiping the dashed T to the middle of the screen, or enter on the full media mode (b) by swiping the dashed T down to the screen.

The user can review and edit every part of the message he is making before he validates it. In this way, he will use the message only when he is satisfied of it. He can validate the message into a challenge or into a response of a challenge by clicking on the appropriate left button of the action bar. Indeed, there should be two items exposed on the action bar while the user is on the message creation fragment: a 'C' item that leads to a challenge creation, and an 'R' item that leads to creating a challenge response.

- If he chooses to use the message to create a challenge, the app redirects him to a list of all his games, so that he can select the game in which he wants to add his challenge.
- If he chooses to use the message to respond to a challenge, the app redirects him to a list of all the games he is playing at, so he can choose the game and then the challenge of the game to which he wants to respond.

The message is saved once validated. The message should be saved in the dual mode if it contains both a media and an email content; in the full media mode if it contains only a media; in the full email mode if it contains only email content, e.g. text and attached files.

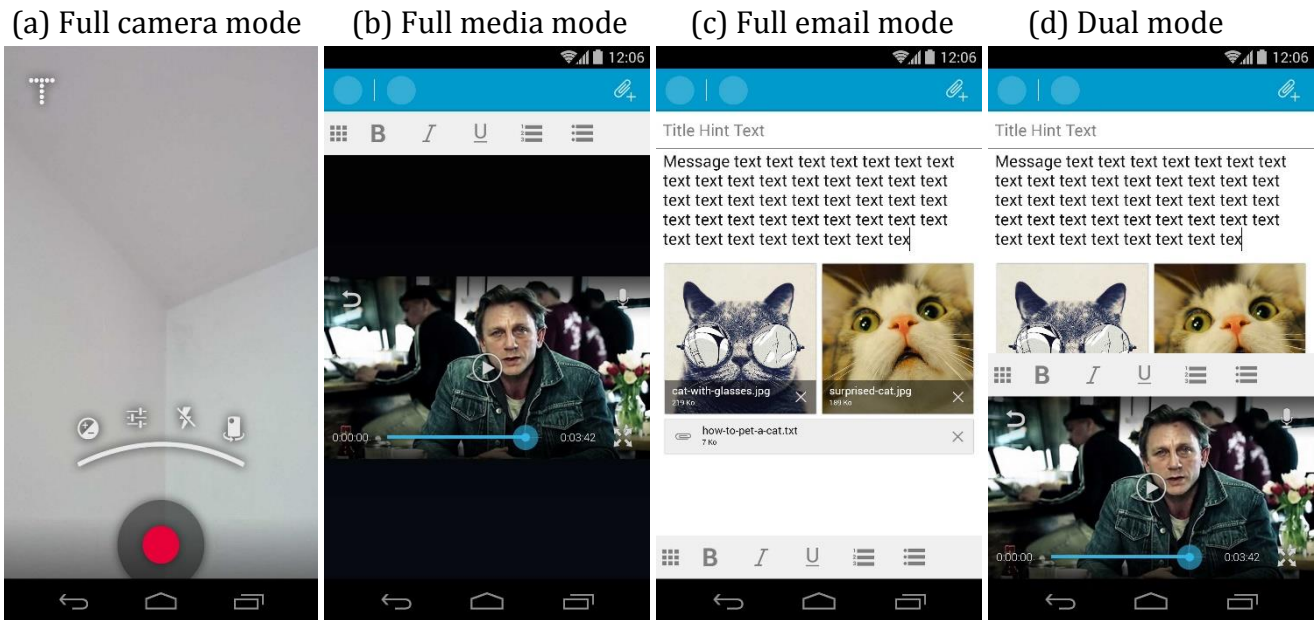


Figure 6 Create Message Modes

Note: The two circles on the left of the action bar should contain a 'C' and an 'R' item to respectively create and respond to a challenge with the underlying message being conceived.

GAME

A user can either play a game or master it. A user should thus be able to have access to two different game lists: the list of the games he is mastering and the list of the games he is playing. From each of these lists, the user has access to the challenges and to the rank of the player of the game he clicked on.

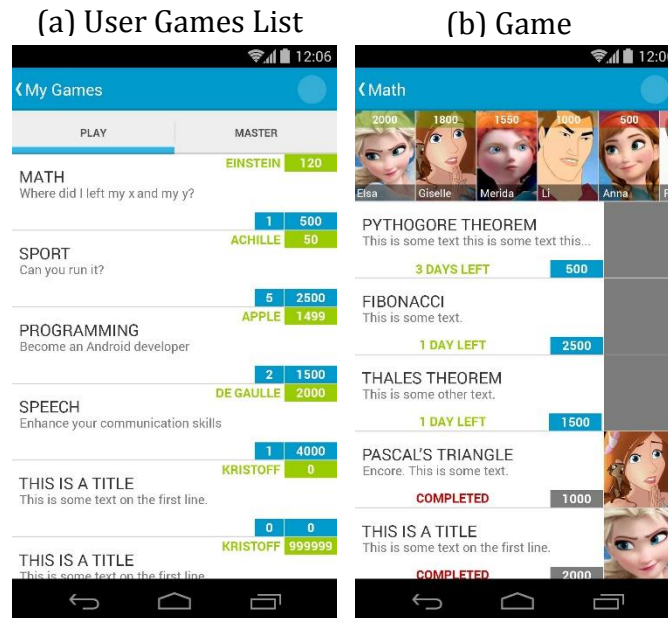


Figure 7 Games and challenges lists

As the picture (a) of Figure 7 shows, a game item contains the title and a short description of a challenge, the player name of the user for the game, the actual score of the game (in green), the number of challenges currently open² and the total score that the user can earn by doing these challenges.

As the picture (b) of Figure 7 shows, a game contains a list of the player ordered in function of their score. Then another fragment contains a list of the challenges ordered so that the most recent challenges are first. A challenge item contains the title challenge, a short description, the number of days left to complete the challenge, the score value of the challenge and a thumbnail of the media. If the challenge was completed, a picture of the winner is displayed instead of the media thumbnail, and when clicking on the item, the user can see the winning message response. For a better understanding of the interaction between the functionalities, the views and the objects of the games, see Figure 8.

² A challenge is closed either when the user already submitted a response to this challenge or when its deadline is passed.

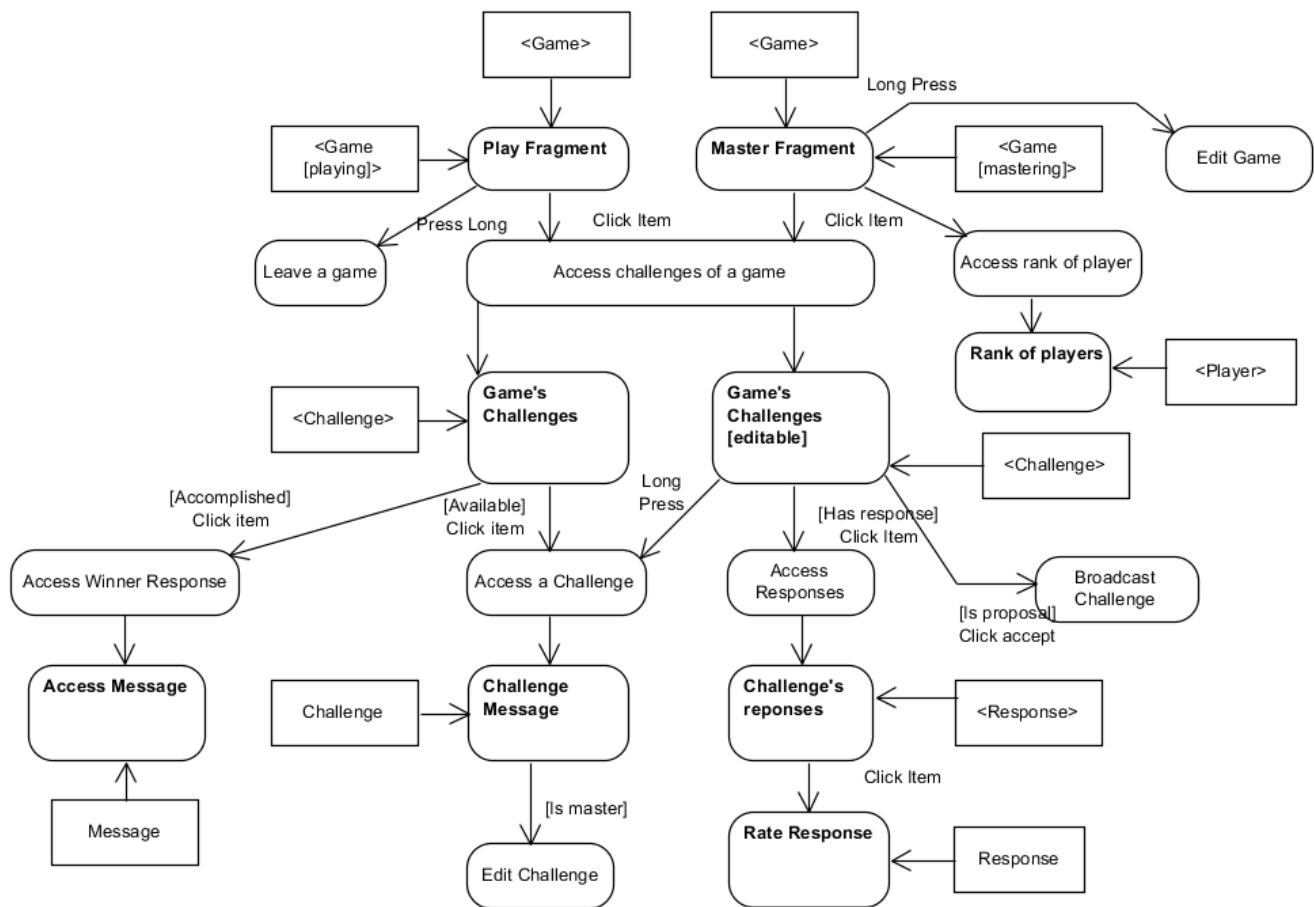


Figure 8 Interaction between functionalities, views and objects of the games

CREATE GAME

As shown in Figures on the right, the user can easily create a game by setting up some basic settings of the game and by selecting players among his friends. For creating a game, the user shall at least add a title to a game and add at least one user. The friends added to the game will automatically become players of the game, and the user creating the game will become the master of the game.

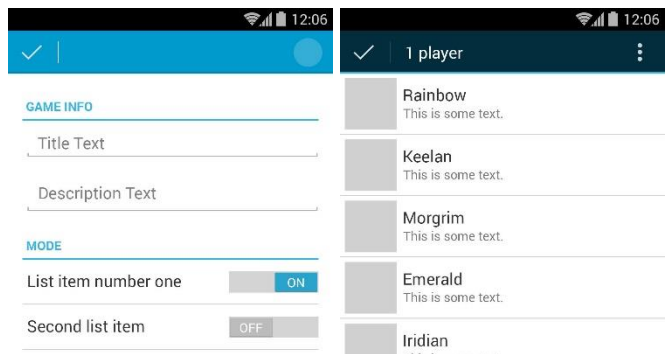


Figure 9 Create Game

MANAGE FRIENDS

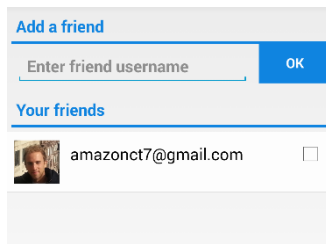


Figure 10 Manage friends

EDIT PROFILE AND SETTINGS

As shown in Figure 11 on the right, the user should easily be able to edit his profile and settings by changing his picture, his username or settings related to the app.

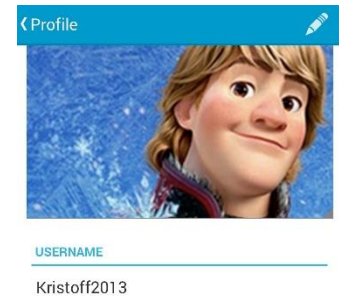


Figure 11 Edit profile and setting

REVIEW CHALLENGE RESPONSE

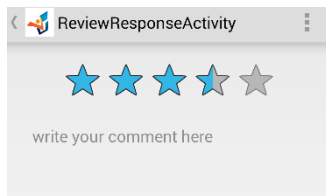


Figure 12 Review Response

As shown in Figure 12 on the left, the master can review the submitted response for a challenge by giving a comment and a rating. The rating can go from zero to five stars.

DESIGN MODEL

This section details the design of **Challengelt** going through the different technologies chosen to implement the system. In particular, this section is focusing on the design of the backend, which is not only critical for the purpose of the app, but also the key part to make the overall system cost effective. Indeed, expect a development fee of twenty-five dollars to publish the app on the Google Play Store, there is no cost associated with deploying an app that does not require a backend. Thus, the cost of the system would mostly come from the backend. The backend of the system relies on Google App Engine (GAE) which was found to be one of the most cost-effective and performant solution. Indeed, not only does the GAE offer a large amount of free quotas and automatically scales to respond to the front-end needs, it is also proven to provide more performant services than its main concurrent Amazon Web Service. For example, Google App Engine has a network twenty times faster, and a latency four times shorter than what Amazon Web Service offers [16].

GOOGLE CLOUD PLATFORM

Google Cloud Platform is a Platform as a Service (PaaS) cloud computing platform for developing and hosting backend of web and mobile applications in the same way that Google uses internally for end-user products like Google Search and YouTube. The Google Cloud Platform is composed of a family of scalable product that automatically allocates more resources for the web application to handle the additional demand. Beyond the services offered by Google App Engine, **Challengelt** will be using the Datastore for storing objects and Google Cloud Endpoint to ensure a REST architecture between the app and the Datastore, and will be using the Cloud Storage to store larger files (pictures, videos and audios).

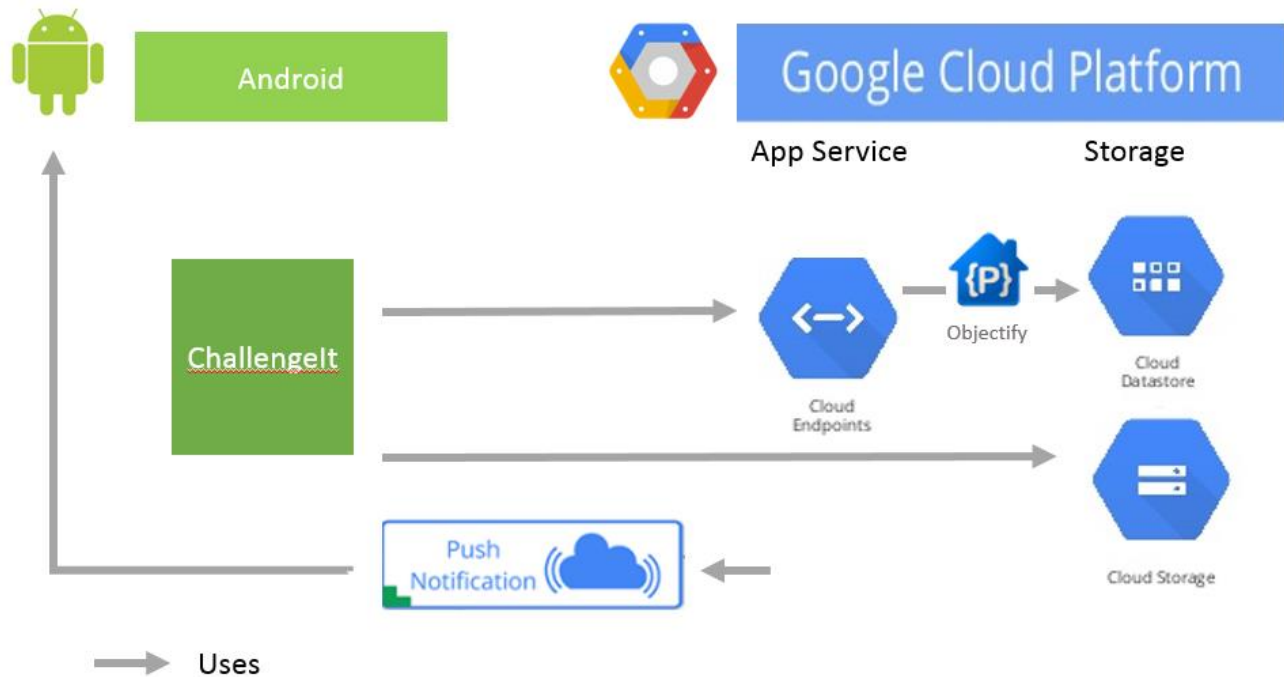


Figure 13 ChallengeIt architecture overview

Note: The services used for ChallengeIt are free up to a certain level of consumed resources, and in the context of this research project, we will not go over these limits.

DATASTORE

Datastore is a highly scalable and highly reliable database that Google offers for storing objects. It is a NoSQL database based on the Big Table technology which is the technology used by Google Research and YouTube. It differs from traditional relational database in many ways. It is less structured database than the RDB but much offer different advantages thanks to its specific features [17]:

- Automated Shredding of the Database, which makes the support for databases on Google servers scalable.
- High Replication, which means data is replicated across multiple datacenters using the optimized Paxos algorithm. It offers time performance on the read and write queries (there is a close server on which the data is) and reliability of the data (no data loss).
- Atomic Transactions, which means that a transaction (which can contain multiple operations) cannot succeed unless every one of its operations succeeds; if any of the operations fails, the transaction is automatically rolled back. This is especially useful for distributed web applications, where multiple users may be accessing or manipulating the same data at the same time.

The Datastore holds data objects known as entities. An entity has one or more properties, named values of one of several supported data types: for instance, a property can be a string, an integer, or a reference to another entity. Each entity is identified by its kind, which categorizes the entity for the purpose of queries, and a key that uniquely identifies it within its kind [17].

Bigtable, which is the underlying structure of the Datastore where the objects are stored, does not support query because there is no column to go through. All entities (rows) of a kind (table) can only be searched on a key. In order to make queries, the Datastore uses an Index Table for properties on which the query should be possible.

One main advantage of the Datastore is its low cost: the amount of free quotas for storage and monthly query is very important and even after those quotas are consumed, the price remains very low.

	Free quota per day	Beyond free quota
Data Stored	1 Go	0,18 USD/Go/mois
Writing operation	50 000	\$0.06/100K operations
Reading operation	50 000	\$0.06/100K operations

Figure 14 Pricing Datastore [18]

With the free quotas and prices shown in Figure 14, the system can easily support an entire small university for almost, if not completely, no cost.

There is then a trade-off between a traditional RDB and the Datastore because if we want to allow all kinds of queries, there is a need to build multiple index tables. Because these index tables can, by term, take a lot of space if we want to allow a large amount of different queries, the cost of the Datastore can be higher than the one of a RDB. However, there is no free quotas for the SQL solution offered by Google and the queries that ChallengeIt is using remain simple. Thus, the Datastore remains for the ChallengeIt the lowest cost solution for an active use with a large amount of users and will remain free in the context of this project research.

To conclude, as many user-ends like Google Search, YouTube, or Snapchat, **ChallengeIt** choice is going toward the Database because of its scalability, high performance and its low cost.

Figure 15 below presents how the ChallengeIt backend database was structured in the Datastore:

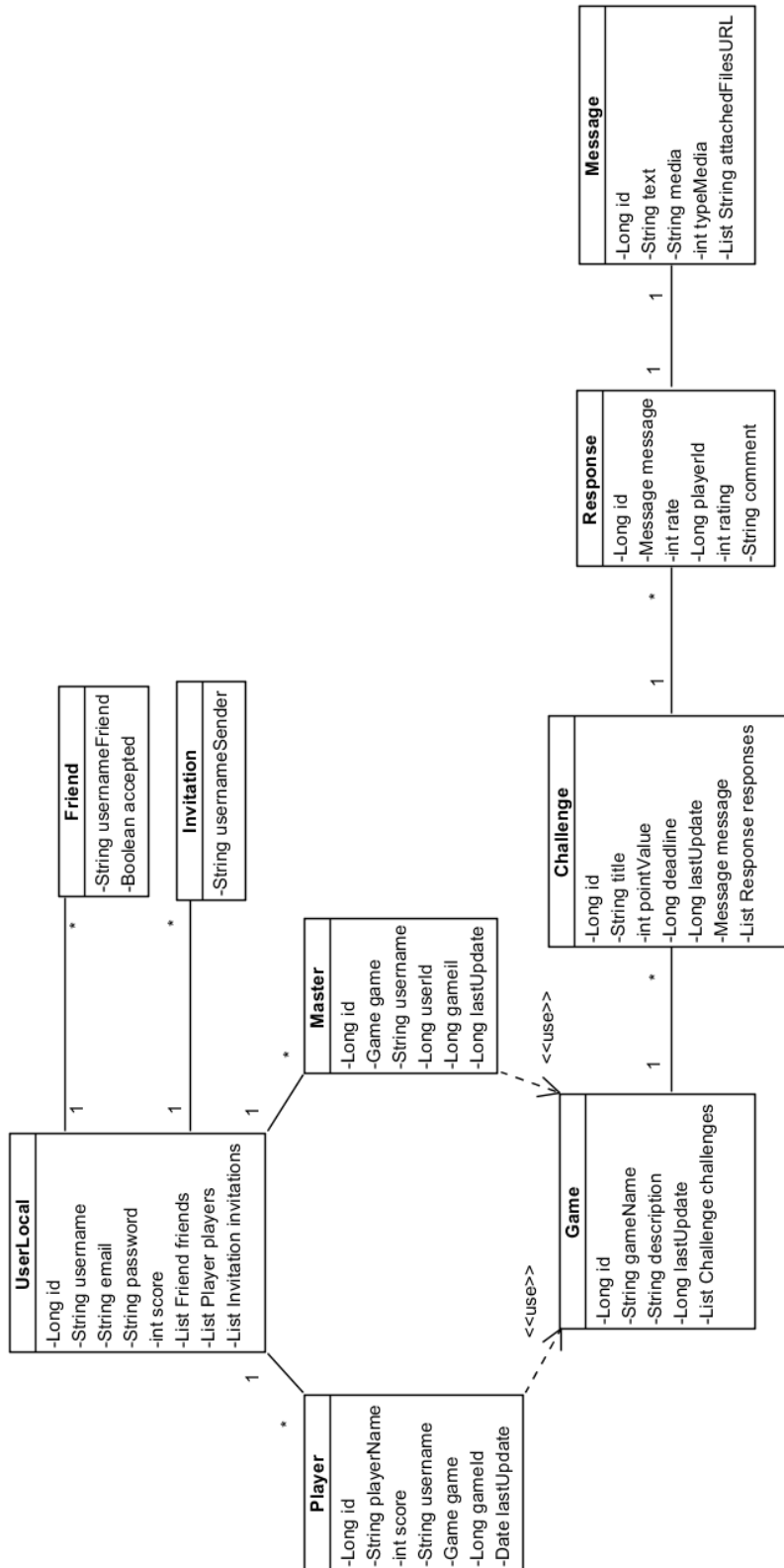


Figure 15 Datastore structure

CLOUD ENDPOINT

Google Cloud Endpoints consists of tools, libraries and capabilities that allow you to generate APIs and client libraries from an App Engine application, referred to as an API backend, to simplify client access to data from other applications. Endpoints makes it easier to create a web backend for web clients and mobile clients such as Android or Apple's iOS.

For mobile developers, Endpoints provides a simple way to develop a shared web backend, and provides critical infrastructures, such as OAuth 2.0 authentication, eliminating a great deal of work that would otherwise be needed. Most importantly, by using GAE for the backend, developers are freed from system admin work, load balancing, scaling, and server maintenance.

It is possible to create mobile clients for App Engine backend without Endpoints. However, using Endpoints makes this process easier because it frees developers from having to write wrappers to handle communication with App Engine. The client libraries generated by Endpoints allow you to simply make direct API calls [19].

Beyond the simplicity and security offered with Google Cloud Endpoint, ChallengeIt is using this technology in order to create REST APIs that will permit to easily support different platforms in the client side in the future like an iOS app or a JavaScript web client.

Figure 16 below shows what using an Endpoints API allows you to do:

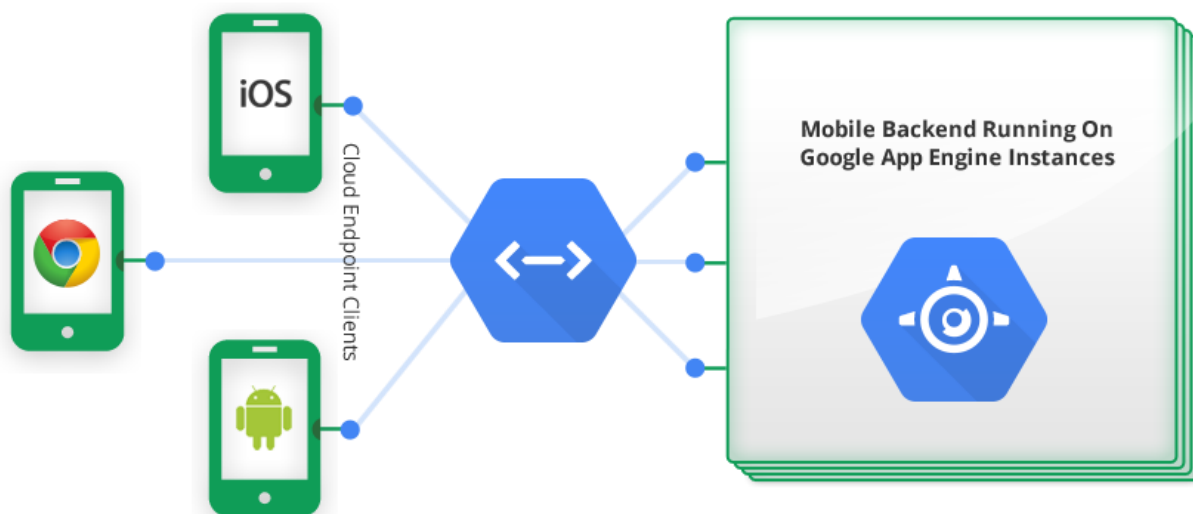


Figure 16 Google cloud endpoint architecture [19]

As shown in Figure 16, the backend performs business logic and other services for Android, iOS, and JavaScript clients. The services offered by the backend are grouped into Cloud Endpoint API. The clients communicate only with the Cloud Endpoints and the Cloud Endpoints communicate with the Google App Engine instance to provide the required services.

ChallengeIt REST APIs can be accessed using the Google App Engine API explorer using the following link https://1-dot-challengeit-92.appspot.com/_ah/api/explorer . For testing purpose, the API will allow call from the explorer as soon as the user is authenticated using a Google Account. When the app is release, the cloud endpoint shall allow only calls coming from authenticated user using the Android OAuth2.0 key, e.g. calls coming from the official **ChallengeIt** Android app. In that way, only users of the app can use the backend for the API calls they are authorized to use, and the data stored on the backend are protected.

Figure 17 and 18 below expose the two REST APIs conceived and implemented for **ChallengeIt**.

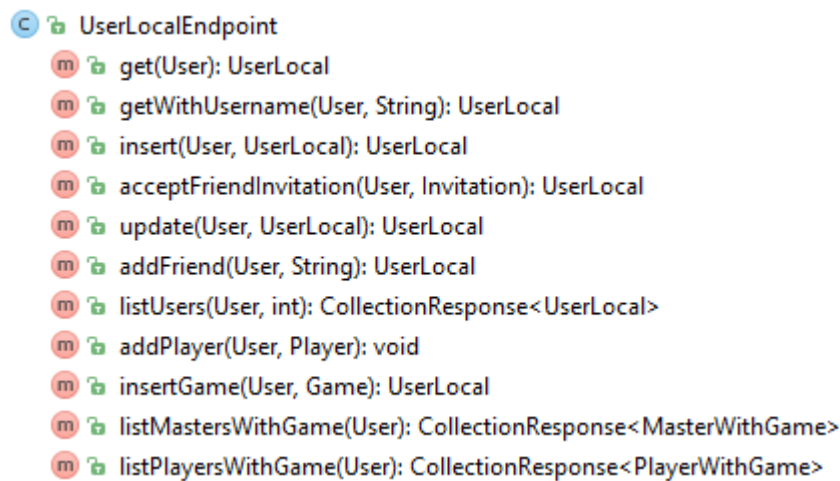


Figure 17 UserLocal API

Note: The user input parameter is an object automatically created on Google Cloud Endpoint from the user authentication. It contains data retrieved from the user Google Account, e.g. in our case the user email address.

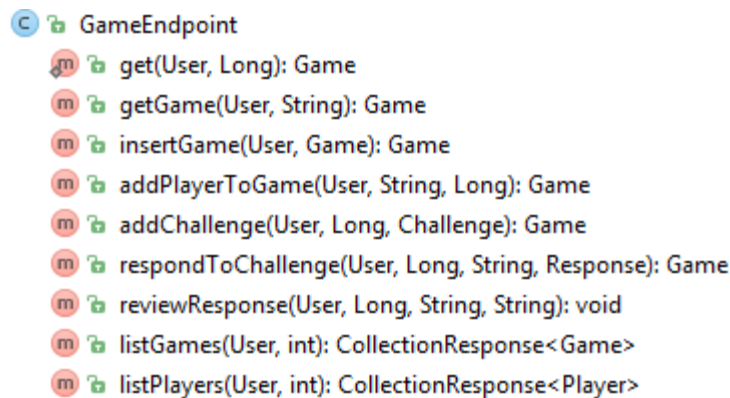


Figure 18 Game API

PROTOTYPE PRESENTATION

This section is presenting the work done on the prototype to give concrete insights of how the app would look like once completed.

CONNECTING TO THE APP

When using the app for the first time, the user need to sign up to the app by selecting one of the Google Account registered on his phone. In the process, **ChallengeIt** asks through Google the permission to the user to use the email address associated with his account. **ChallengeIt** saves the choice made on the app-shared preferences (local memory reserved for the app). Thus, when connecting again to the app, the user will automatically be signed in on the app using the same account that he had selected. Meanwhile, on the sign up process, the app makes a call on the backend for a creating a user entity using the email address of the Google Account as the identifier of the entity. In this way, the backend is able to authorize or not users to make API calls. For example, only a user authenticated and authorized is able modify his profile. It guarantees that the app is secure and that sensitive data are protected.

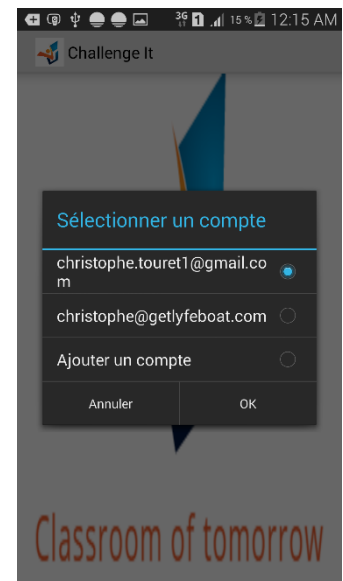


Figure 19 Sign up with Google Account



Figure 20 Connection to the app

For next uses of the app, the user is automatically signed-in into the app when launching it. The app makes a call on the backend to retrieve the user entity previously created on the sign up process. Every calls made while using the app are authenticated and authorized with the user google credentials.

MAIN ACTIVITY

When connected the user have access to the main activity with the following four fragments.

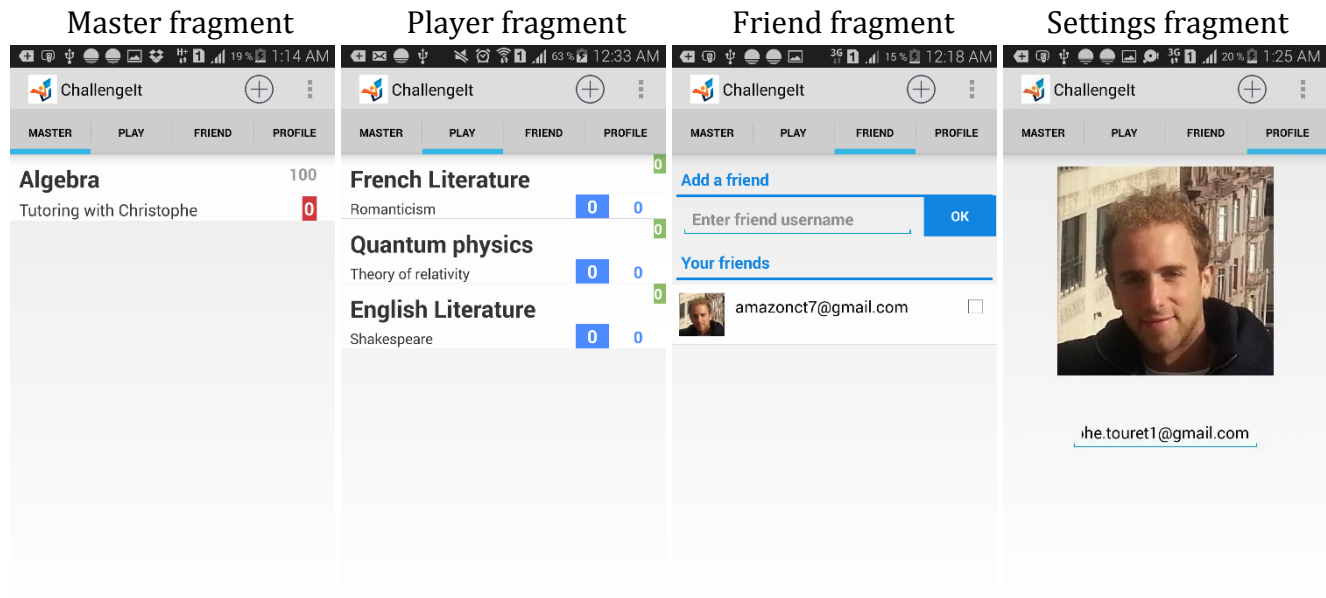


Figure 21 Main activity fragments

The main activity opens first on the play fragment. The user can navigate between the four fragments presented above by swiping or clicking on the header of the tabs.

MASTER

This section is presenting the different views made for a master.

Master Game Activity

In the master fragment, the user has access to the games he has created. When clicking on a game, the user access to the list of challenge and to the rank of players of the game. The user can create challenges and edit them using the item 'edit' of the action bar.

In the example on the right, the game “Algebra” has one challenge with the title “Fundamental Theorem”, it has a short description “Demonstrate the corollary 2”, it is due in 3 days and it worth 1 000 points. Only one player is playing this game, which is the one represented by the picture on the top left of the activity.

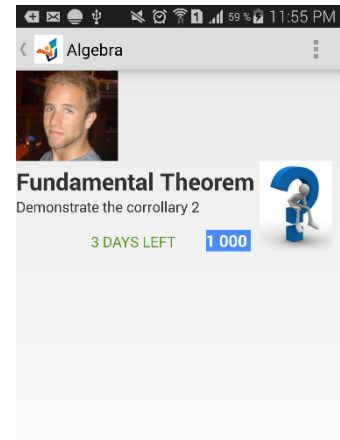
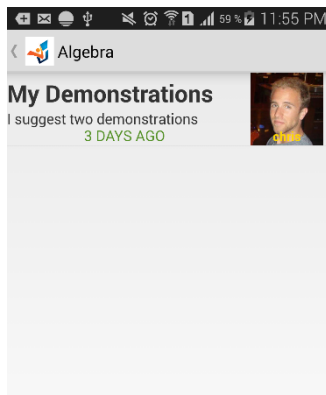


Figure 22 Master Game Activity

List Responses Activity

When clicking on a challenge, the master accesses to the list of responses done by the players of the game.

In the example on the left, the player “Chris” has responded to the challenge. The response he made has “my demonstrations” as a title, and the description is “I suggest two demonstrations”. The response was submitted 3 days ago.



Response Activity

When clicking on the response, the master accesses to the content of the response on the response activity.

The master can then click on the review button on the action bar to review the response, e.g. rate and comment it as show in Figure 24.

Figure 23 List Responses Activity

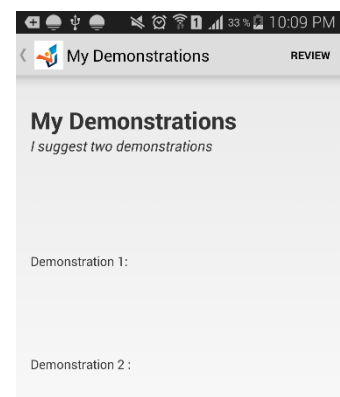


Figure 24 Response activity

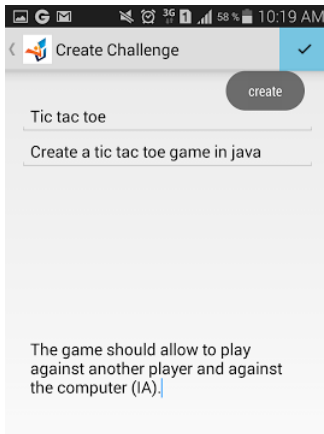


Figure 25 Create Challenge Activity

Create Challenge Activity

When clicking on the action item “create challenge” on the master game activity, the master accesses to the create challenge activity.

For creating a challenge, the master needs at least to enter a title. In the example on the left, the title is “Tic tac toe”, the description is “create a tic tac toe game in java” and the content is what is below that.

PLAYER

This section is presenting the different views made for a player.

Player Game Activity

In the player fragment, the user has access to the games he has been invited into. When clicking on a game the user access to the list of challenges as well as the rank of user.

In the example on the right, the game “English Literature” has one challenge with the title “Comment Quote”, it has a short description “Comment the quote”, it is due in 3 days and it is worth 1 000 points. Only one player is playing this game, which is the one represented by the picture on the top left of the activity.

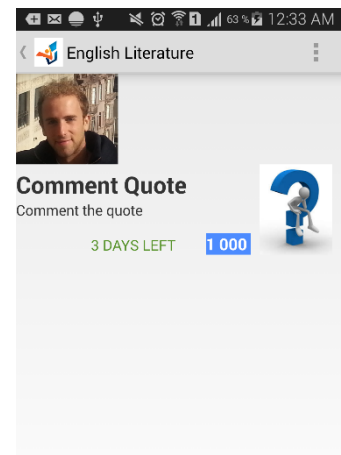


Figure 26 Player Game Activity

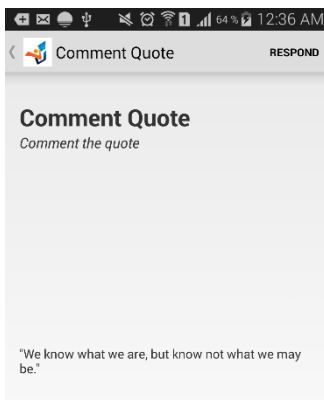


Figure 27 Challenge Activity

Challenge Activity

When clicking on a challenge, a player accesses the content of the challenge on the Challenge Activity. As an example, Figure 27 shows the content of the challenge “Comment Quote” previously used on the Challenge Activity.

The player can then click on the respond button to create and submit a response for the challenge.

Create Response Activity

When clicking on the response button, the user can create a response on the create response activity as shown in Figure 28. In this example, the title of the response is “My comment”, the short description is “Inspired from the comments of Gabriel Rieger”, and the content of the response is what follow below that.

Once he is done with the response message, the user validate it to and the app submit the response on the challenge.

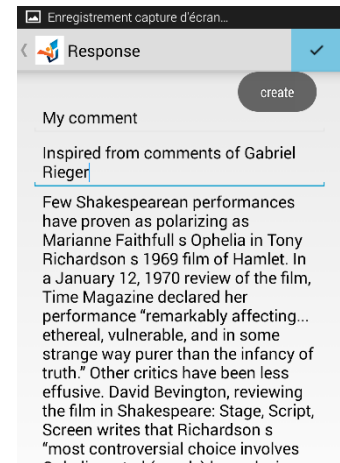


Figure 28 Create Response Activity

FRIEND

This section is describing how a user can manage friends with the friend fragment.

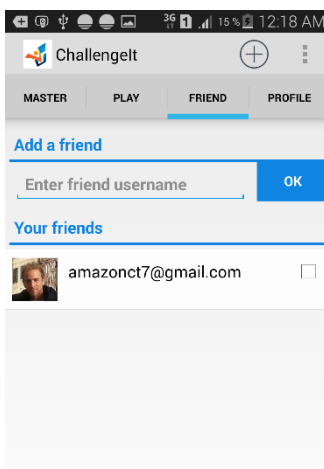


Figure 29 Friend fragment

A user can add a friend by entering the username of user he wants to invite as a friend. If the username does not exist, the app displays a toast message explaining that the username could not be found. Otherwise, the friend is added to the list of friends, and the friends list fragment is automatically refreshed. When making a call to the backend to add a friend, the UserLocal Endpoint is checking that the friend was not already added into the user friend list. In the completed app, the friend request shall be accepted before the user is added as a friend to the list. See Figure 32 “Add a friend sequence diagram” in the appendix, to understand the friend request process that was implemented on the backend.

CREATE GAME

Using the action bar, the user can create a new game by clicking on the icon “new game”.

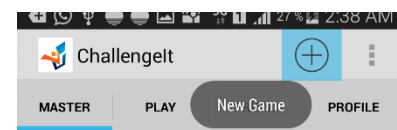


Figure 30 New game item

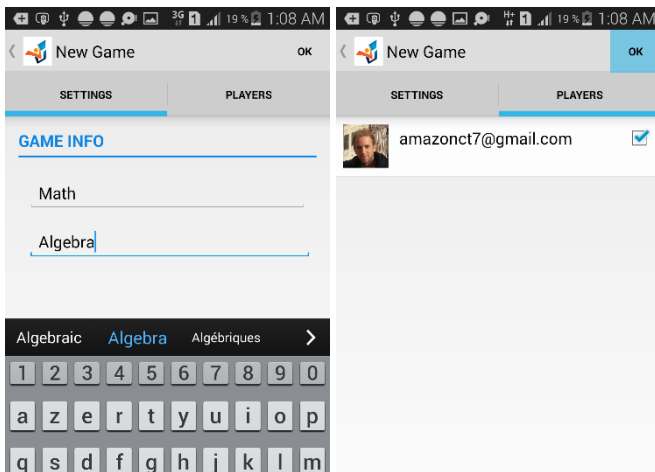


Figure 31 Create New Game Activity

The user needs to add at least a title and a player to the game to be able to create it. The user can optionally add a description to the challenge.

The user can then validate the game by clicking on the “OK” button on the action bar.

See Figure 33 “Add a player into a game sequence diagram” in the Appendix to see the process involved with adding a player into a game.

FUTURE WORK

This section is presenting the list of features that need to be implemented to complete the project ChallengeIt from the prototype:

- Add points and deadline on a Challenge.
- Review a response.
- Show horizontal rank of player with their accumulated points on the game.
- Implement the point system:
 - Add points to a player after the master reviewed his response.
 - Show the points accumulated on a game for each player.
- Allow a user to quit a game.
- Refresh the list of master games when the user created a game.
- Implement the GCM for real time refreshing of the data.
- Implement friend request system in the frontend (already implemented in the backend):
 - Show friend requests received in the friend fragment.
 - Allow a user to accept a friend request.
- Allow a master to edit the challenges and the game settings of the game he is monitoring.
- Allow a user to create a message using a combination of media, text and attached files.
- Allow a user to directly create a message on the Main Activity and then decide to create or respond to a challenge with it.
- Allow a user to logout.
- Create a game with public and private mode.
- Allow a user to add and edit a picture profile.
- Show local notifications when the user has new friend requests, new challenges, new responses or new games.

DISCUSSION AND CONCLUSION

This report discussed the objectives, requirements, ergonomics, design, and implementation details of a novel game-based learning environment, **ChallengeIt**, which is deployed on Android devices.

ChallengeIt provides an innovative solution as it represents a fusion of the edutainment game genre and the interactive classroom tool. It is non-proprietary, economical to implement/use, easy to use, and provides convenient access to learning activities wherever the student is at any time. **ChallengeIt** also represents a departure from previous attempts to implement game-based digital classrooms.

At this moment, the backend and the frontend prototype presented on this report are completed. Once the product will be completely implemented and tested, it would be interesting to use it in ERAU classrooms to assess its usability and performance.

ChallengeIt is aiming to utilize technology that most students already have by creating an interactive and fun environment to further studies both within and outside the classroom. By presenting students with course-related “challenges” by their instructors, students would compete between each other for the most number of correct answers, and be motivated to learn new concepts outside the normal course requirements to be able to complete them.

APPENDIX

SEQUENCE DIAGRAMS

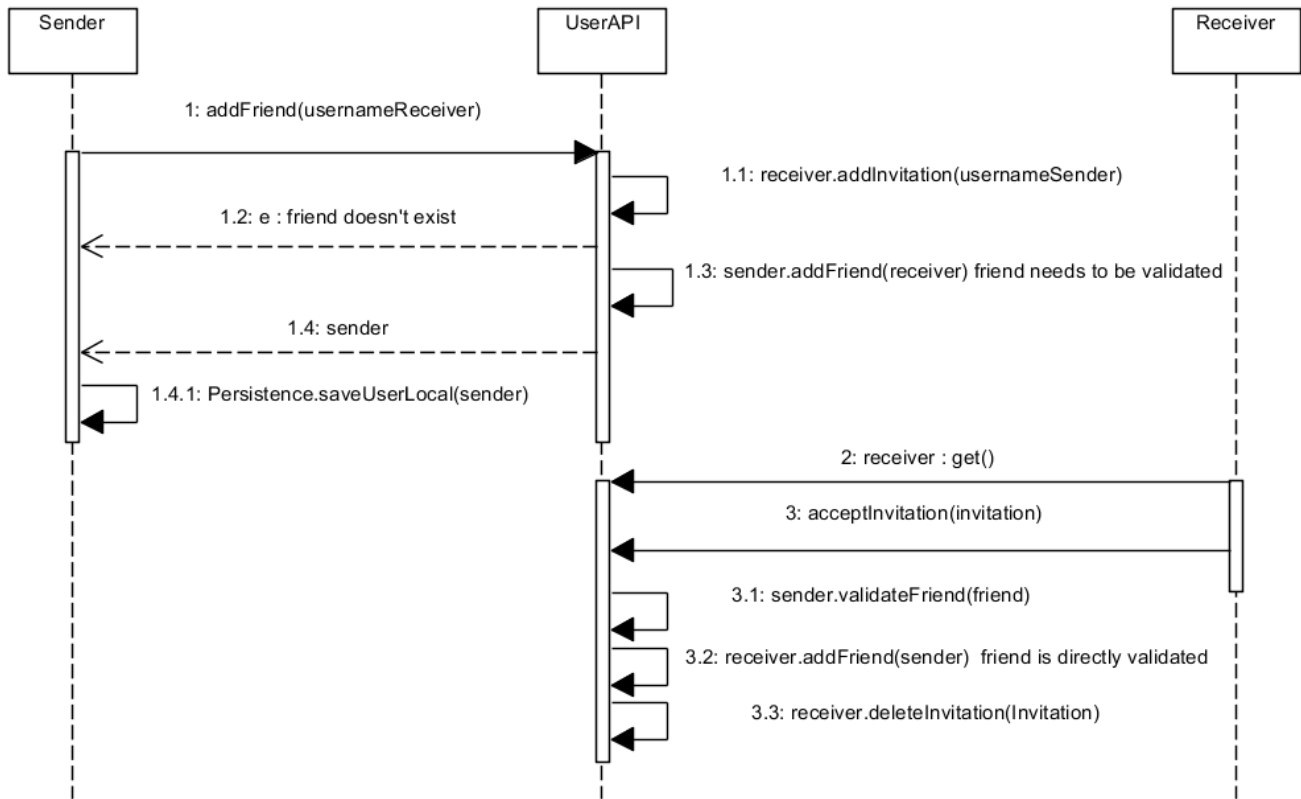


Figure 32 Add a friend sequence diagram

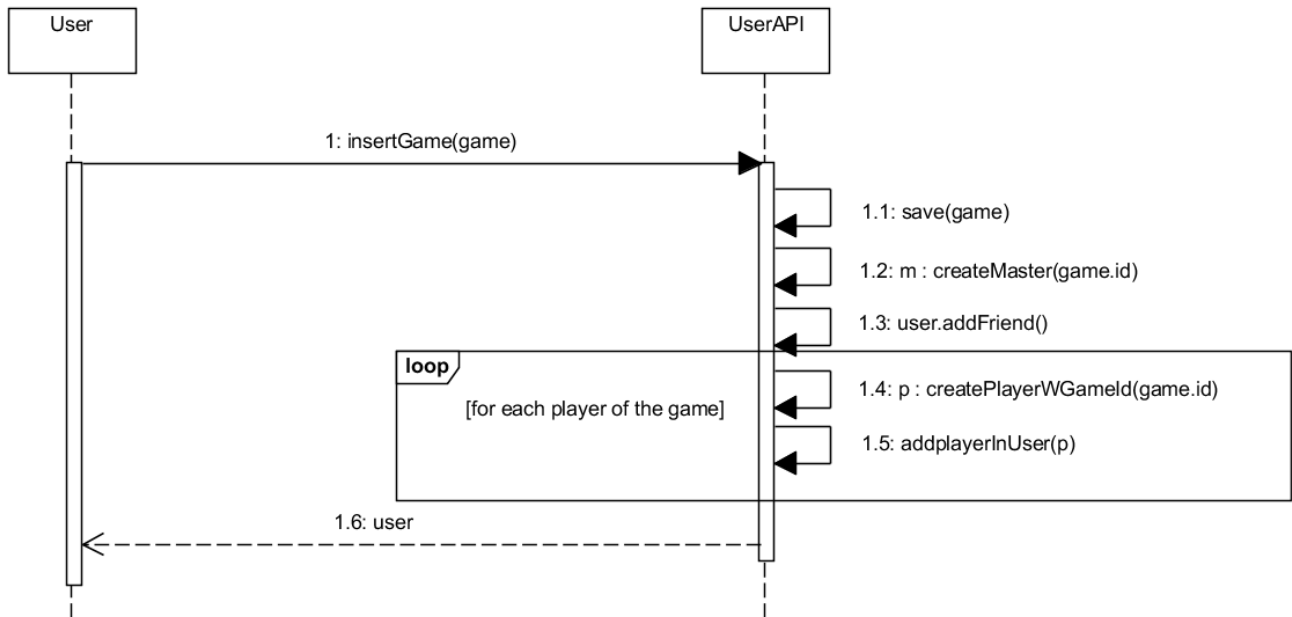


Figure 33 Add a game sequence diagram

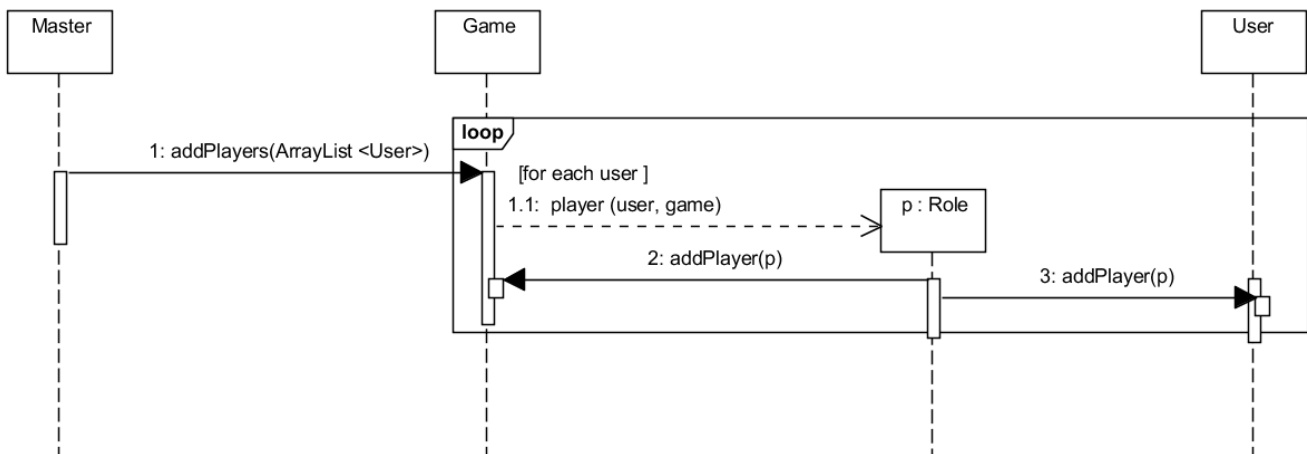


Figure 34 Add a player sequence diagram

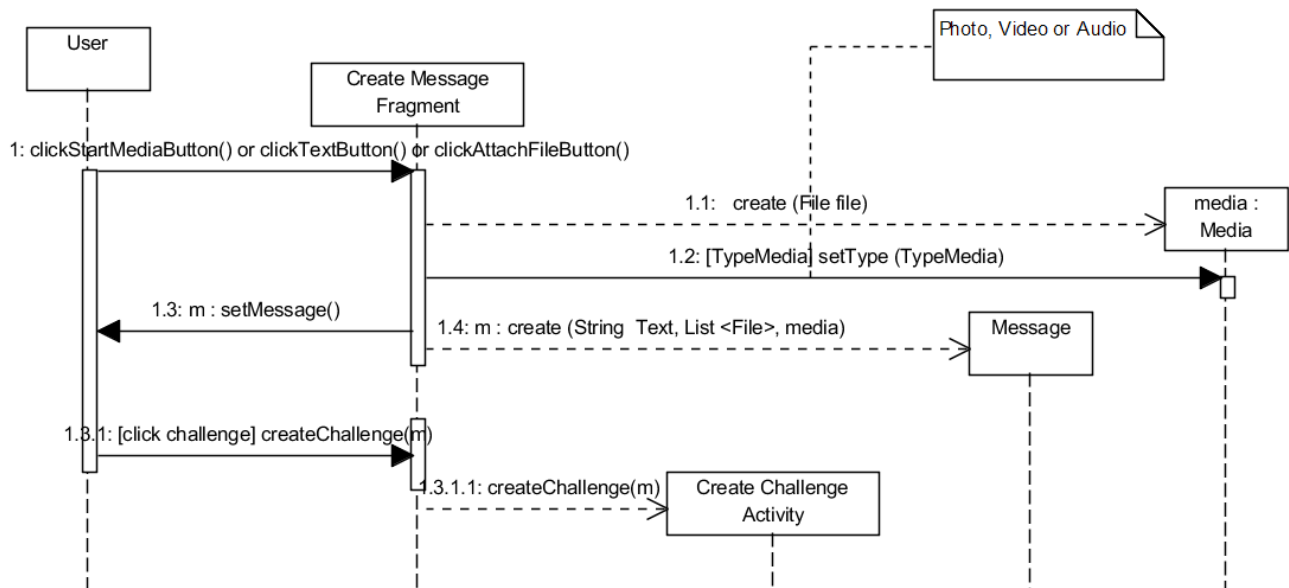


Figure 35 Create message sequence diagram

REFERENCES

- [1] A. Deal, "Classroom Response Systems," *Carnegie Mellon*, 2007.
- [2] Qualcomm, "WE Learn: Building the 21st Century classroom," *Qualcomm Wireless Reach*, 2014.
- [3] E. A. House, "The High Cost of Wisconsin's Dropout Rate," *Friedman Foundation for Educational Choice*, 2009.
- [4] J. M. Reaynol Junco, *Connecting to the Net.generation*, 2007.
- [5] seriousgames.net, "Serious games interactive," [Online]. Available: <http://www.seriousgames.net/>.
- [6] A. J. Stapleton, "Serious Games: Serious Opportunities," *Australian Game developers' Conference*, 2004.
- [7] A. a. T. I. o. E. (IDATE), "Serious Game, un chiffre d'affaires mondial de 10 milliard EUR en 2015," Oigiworlworld by IDATE, 2010. [Online]. Available: http://www.idate.org/fr/Actualites/Serious-Games_643.html.
- [8] "Gamification Programs Engage Healthcare Consumers," Smith&Jones Healthcare Marketing. [Online].
- [9] K. Y. Cathy Ringstaff, "Integrating Technology into Classroom Instruction," *Apple classroom of tomorrow*, 1996.
- [10] D. M. Nussbaum, "The classroom of tomorrow built with today's technology," *Microsoft Research*, 2006.
- [11] D. Sessoms, "Interactive Instruction: Creating Interactive Learning Environments Through Tomorrow's Instructors," *International Journal of Technology in Teaching and Learning*, 2008.
- [12] Microsoft, "Microsft Interactive Classroom," Microsoft, [Online]. Available: <http://www.pil-network.com/Resources/Tools/Details/da5d09fd-9362-43b6-9e96-cf9aad66e6c>.
- [13] Apple, "Apps in the Classroom : Using iOS Apps for Teaching and Learning."
- [14] C. S. L. Bilal Younts, "Integrating Serious Games in Higher Education Programs," *Academic Colloquium 2010: Building Partnership in Teaching Excellence*, 2010.
- [15] P. J. Summaris Roodt, "Evaluating Serious Games in Higher Education: A Theory-based Evaluation of IBMs Innov8," 2008.
- [16] S. Stadil, "By the numbers: How Google Compute Engine stacks up to Amazon EC2," Gigaom, 2013. [Online]. Available: <https://gigaom.com/2013/03/15/by-the-numbers-how-google-compute-engine-stacks-up-to-amazon-ec2/>.
- [17] G. a. engine, "Java Datastore API," 2013. [Online]. Available: <https://developers.google.com/appengine/docs/java/datastore/>.

- [18] G. Cloud, "Datastore," [Online]. Available: <https://cloud.google.com/datastore/>.
- [19] C. Google, "Google Cloud Endpoint Overview," Google, [Online]. Available: <https://cloud.google.com/appengine/docs/java/endpoints/>.
- [20] J. Foreman, "Next-generation Educational Technology versus the lecture," *Educause review*, 2003.
- [21] d. l. s. e. d. l. r. Ministère de l'éducation nationale, "Jeux sérieux, mondes virtuels," eduscol, [Online]. Available: <http://eduscol.education.fr/numerique/dossier/apprendre/jeuxserieux/notion/typologie>.
- [22] B. Boehm, "Spiral Development: Experience, Principles, and Refinements.," 2000.
- [23] J. S. Ken Schwaber, "The Scrum Guide," wikipedia, 2013. [Online]. Available: [http://en.wikipedia.org/wiki/Scrum_\(software_development\)](http://en.wikipedia.org/wiki/Scrum_(software_development)).