KADİR HAS UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

USING GAMIFICATION APPROACH FOR SCORING APPLICATION DEVELOPERS OF AN APAAS

GRADUATE THESIS

MUSTAFA VOLKAN GÜROL

AUGUST 2014

MUSTAFA VOLKAN GÜROL

Master Thesis

2014

USING GAMIFICATION APPROACH FOR SCORING APPLICATION DEVELOPERS OF AN APAAS

MUSTAFA VOLKAN GÜROL

B.S., Statistics and Computer Science Kadir Has University, 2011
B.S., Computer Engineering, Kadir Has University, 2011
M.S., Computer Engineering, Kadir Has University, 2014

Submitted to the Graduate School of Science and Engineering
in partial fulfillment of the requirements for the degree of
Master of Science
in
Computer Engineering

KADİR HAS UNIVERSITY
AUGUST, 2014

KADİR HAS UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

USING GAMIFICATION APPROACH FOR SCORING APPLICATION DEVELOPERS OF AN APAAS

MUSTAFA VOLKAN GÜROL

APPROVED BY:

Mehmet N. Aydın	Assoc. Prof.
A. Selçuk Öğrenci	Asst. Prof.
N. Ziya Perdahçı	Asst. Prof.

APPROVAL DATE: 28/08/2014

“I, MUSTAFA VOLKAN GÜROL, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.”

MUSTAFA VOLKAN GÜROL

USING GAMIFICATION APPROACH FOR SCORING APPLICATION DEVELOPERS OF AN APAAS

Abstract

Cloud Computing has experienced a predominant growth during the past years and businesses who have shifted to the cloud model acknowledge the benefits, however there is still a room for expansion globally, and the growth is expected to continue. IT has invented new ways of accelerating application (app) development during the past 100 years. Cloud is no exception. Platform as a Service (PaaS) is, emerging to raise the abstraction bar of operational aspects of like deployment, availability, scalability, security. Application Platform (aPaaS) is emerging to raise the abstraction bar of app development, aiming to make app development faster and easier for developers. APaaS are offering companies decreased development costs, and higher time to market while boosting productivity. Despite all the aforementioned advantages, to attract attention of app development enthusiasts in a global scale is a challenge. An app platform should be engaging, easy to understand, easy to use, as well as rewarding. Developers should enjoy working with it; the platform should exert the so-called coolness factor, so that a developer sticks to it until the app is leveraged by its customer.

Our main focus on this thesis is, with the help of gamification techniques, to convert an aPaaS into an intuitive, engaging platform for developers. We chose ImonaCloud (imona.com) as the application platform. In the scope of this thesis, 30 developers scored and rewarded according to their actions and achievements on the platform. By looking their meta-data actions and scores they got, it can be said that, our scoring model is viable for that kind of aPaaS platforms. This model may help not only the developers but for the novice programmers to learn the platform faster and become more engaged.

Keywords: Cloud, Gamification, Scoring, Meta-data, aPaaS, Developer experience

BİR APAAS ÜZERİNDE UYGULAMA GELİŞTİRMECİLERİN OYUNLAŞTIRMA YAKLAŞIMI İLE SKORLANMASI

Özet

Bulut bilişim son zamanlarda etkin bir büyüme sergiliyor ve çalışmalarında bulut modelini tercih eden işletmeler elde ettikleri avantajın farkındalar, ne var ki uluslararası ortamda halen büyüme ve gelişme için yer var ve büyümenin devam etmesi bekleniyor. Bilişim Teknolojileri son asır içerisinde hızlandırılmış uygulama geliştirmenin yeni metotlarını keşfetmeye başladı. Bulut bilişim de buna istisna değil. Platformlar (PaaS) deployment, erişebilirlik, boyutlandırma ve güvenlik gibi işlevsel bakımlardan sanal bir çizgide yükseliyor. Uygulama Platformları (aPaaS) ise uygulama geliştirme, uygulama üretiminin geliştirici için hızlanması ve kolaylaşması konularında yükseliş gösteriyor. aPaaS işletmelere düşük maliyetli, üretkenliği arttırırken pazarda daha uzun süre kalabilme imkânları sunmaktadır. Bütün bu belirtilen avantajlara rağmen küresel uygulama geliştirme odağını çekmeyi başarmak başlı başına bir mücadeledir. Bir uygulama geliştirme platformu çekici olmalıdır, kolayca anlaşılabilmelidir, kolay kullanılabilmelidir ve ödüllendirici olmalıdır. Geliştiriciler onunla çalışmayı eğlenceli bulmalı, müşterisinin ihtiyacını karşılayana kadar platformla arasında bağ kurmalıdır.

Bu tezin asıl amacı, oyunlaştırma tekniğinin de yardımı alınarak, uygulama PaaS platformunu geliştiriciler için güçlü ve çekici hale getirmektir. Biz, Imonacloud’u (imona.com) uygulama platformumuz olarak tercih ettik. Bu tez kapsamında 30 geliştirici bu platform üzerinde yaptıkları işlemler ve kazandıkları başarılara göre puanlandılar ve ödüllendirildiler. Yaptıkları işlemlere ve kazandıkları puanlara bakıldığında, skorlama modelimizin bu tip PaaS platformlarda yapılabilirliği söylenebilir. Bu skorlama modeli, yalnızca geliştiriciler için değil, programlamaya yeni başlayanlar için de platformun hızlı bir şekilde öğrenilmesinde ve bağlanılmasında yardımcı bir unsur olabilir.

Anahtar Kelimeler: Bulut, Oyunlaştırma, Skorlama, Meta-data, aPaaS, Geliştirici deneyimi

Acknowledgements

I would like to thank to my thesis supervisor Mehmet N. Aydın and my manager Bahadır Baran Ödevci who has provided their endless support throughout this work.

I want to thank my family for their love and supports throughout my life. This journey would not have been possible without them.

Lastly, I would like to sincerely thank my all teachers and friends.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

aPaaS	Application Platform as a Service
iPaaS	Integration Platform as a Service
SaaS	Software as a Service
IaaS	Infrastructure as a Service
SME	Small & Medium Enterprise

Chapter 1

Introduction

Application platforms provide a new paradigm for developing enterprise software. Abstraction level is raised for developers to build web and mobile-based software solutions. Speed and ease of building software is complemented with model driven approach. This is innovation in today's ever growing app economy.

However it is still a challenge for a developer to make use of an approach to follow to succeed in this new paradigm. If they cannot find their way on the development platform soon after they become engaged, they might become increasingly disengaged and frustrated during learning process. It is too hard or they don't know where to start or go next. In this thesis we adopt a gamification approach as a solution of engaging developers who develop business apps on an aPaaS platform. We chose ImonaCloud to showcase the model we developed using gamification.

"I'm on a Cloud" (imona.com) is an aPaaS (application platform as a service) developed by Imona Technologies Ltd. A web based platform that makes it easier and faster to develop business apps.

ImonaCloud platform provides several features both for the app developers and end-users: reporting, dashboard of gauges, task and process management, user management, scheduling, audit trail and so on. On the other hand, some features like development editor are provided exclusively for the platform developers. They can define pages for their apps easily just by dragging and dropping the fields on the screen and writing java-based scripts. Application developers don't have to master on programming, but still can build quality, mobile ready, scalable, multi-tenant business apps.

The apps developed in the platform can be deployed to Imona Marketplace, and be marketed to small and medium enterprises in pay per use basis.

In recent years, gamification approach has been adopted in different contexts, for instance, enterprise area. According to Zichermann [1], gamification is the use of game play thinking and mechanics to solve problems and engage audiences. Furthermore, [2] defines gamificaion as “it is the use of game design elements in the non-game context”.

In this research project, we integrated game mechanics like points, level, badges and leaderboards on ImonaCloud platform aiming to encourage developers to improve their engagement and effectiveness on their platform practices. Additionally one may consider gamification for its affects on:

- Attracting more customers and developers to cloud platform in order to increase platform’s site traffic.
- Matching companies with developers so that employers find the right developers for their custom requirements.

This is subject to further investigation as a future research topic.

1.1 Thesis Structure

The thesis has five main parts:

- Chapter 1 Introduction,
- Chapter 2 Related Studies,
- Chapter 3 Model Proposed,
- Chapter 4 Data Analysis, Findings and Discussion,
- Chapter 5 Conclusion and suggestion for further research.

In the introduction section, the problem statement, the purpose and objectives of the thesis were described. In the second chapter relevant studies and researches on that thesis were analyzed. Researches focused deeply on three main topics: gamification, cloud computing and aPaaS. In the third chapter, gamification model is proposed. Three game elements were defined: Points, Badges and Level System. Chapter 4 focuses on analyzing pilot study of gamifying ImonaCloud platform. In the first instance, possible valuable meta-data actions on that platform were determined. Afterwards, developers' scores and their effectiveness were investigated. Presented results were discussed. The final chapter provides the conclusion and suggestions for the future researches on that thesis.

Chapter 2

Related Studies

2.1 Gamification Studies

As Gartner predicts [3], by 2015, 40% of Global 1000 organizations will use gamification as the primary mechanism to transform business operations. Many popular websites are already experimenting gamification techniques as a way to boost engagement. Furthermore, there is an increasing number of successful startup companies whose entire service is focused on adding a gamified layer to a core activity [4]. In this section, we'll examine gamification use cases.

According to Deloitte Company researches, Engine Yard, a Platform as a Service (PaaS) company, built a self-help community portal to improve customer support. To encourage regular use, customers can earn achievements for specific behaviors, complete missions to win rewards and badges, and level-up with accumulated experience [5].

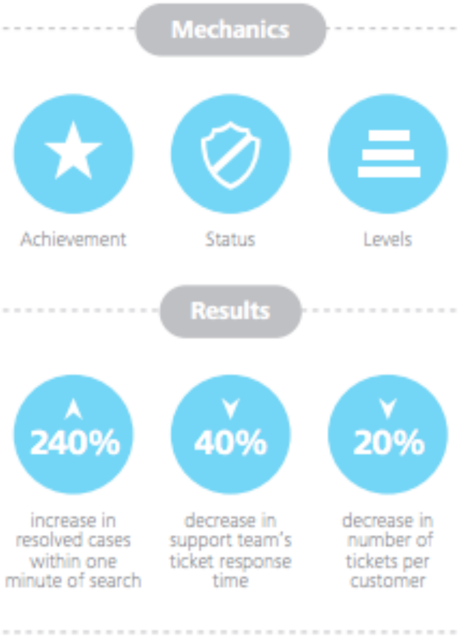

Mechanics
Achievement
Status
Levels
Results
240%
increase in resolved cases within one minute of search
40%
decrease in support team's ticket response time
20%
decrease in number of tickets per customer

Figure1: B2C Case Study: Engine Yard [5]

Marketo, a marketing automation software company, launched Marketo Community to connect customers and partners and foster exchange of expertise. Users can earn points and badges for community engagement, contributions and accelerated onboarding, and can achieve statuses and earn rewards [5].

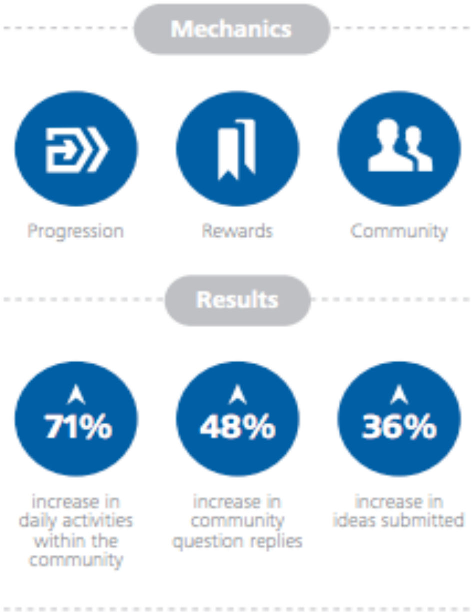

Mechanics
Progression
Rewards
Community
Results
71%
increase in daily activities within the community
48%
increase in community question replies
36%
increase in ideas submitted

Figure2: B2B Case Study: Marketo [5]

Box, a cloud-based file sharing platform, gamified an employee conference to avoid presentation fatigue by creating a platform-neutral website. Participants used mobile devices to answer training questions and complete networking challenges, and points were displayed on a real-time leaderboard [5].

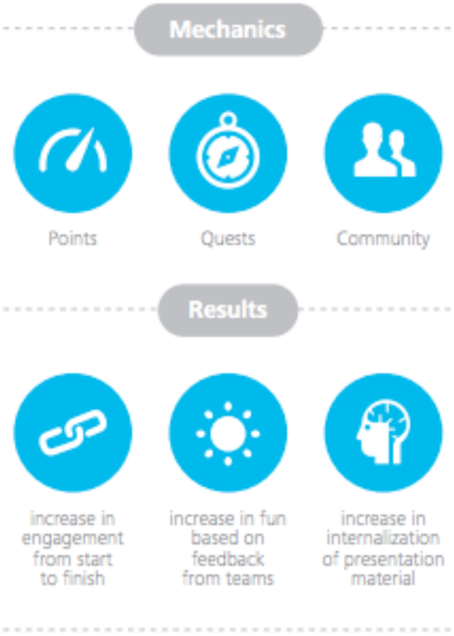
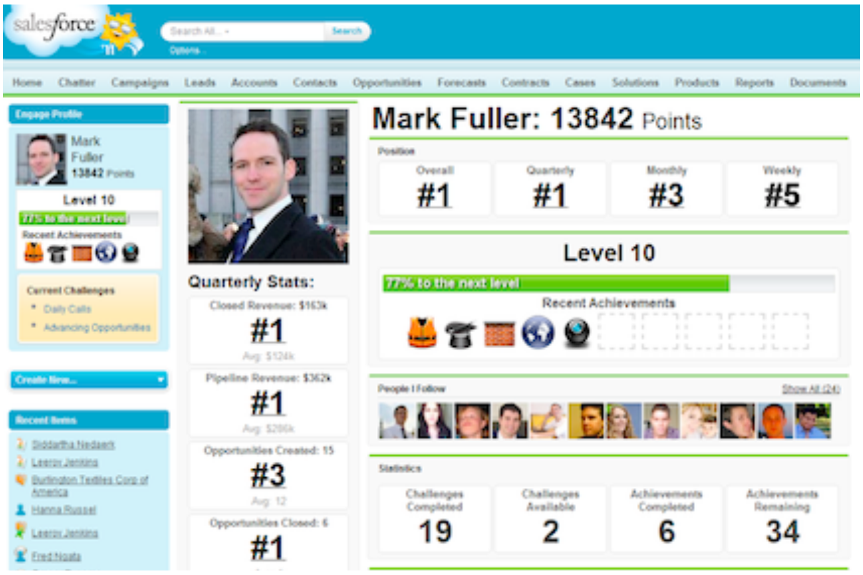

Mechanics
Points
Quests
Community
Results
increase in engagement from start to finish
increase in fun based on feedback from teams
increase in internalization of presentation material

Figure3: B2E Case Study: Box [5]

The next example also aims to increase employee performance by leveraging game mechanics. Deloitte Learning Academy (referred as DLA) has a rich portfolio of content types in which they provide programs for digital executive development education. The problem with executives tends to be that their busy business life does not make taking the advantage of professional development content an easy job. Their business life is built on competing and leaving personal training to inferior. However DLA still managed to implement a system which utilized the concepts of points, achievements, and missions to draw executives into learning activities [6]. A series of achievements and earning points for their accomplishments are put on display on their profile. To do this, they give missions that lead them into learning path, even taking advantage of leaderboard to enhance the competition experience in several key areas. DLA's gamification efforts have bloomed in mere 3 months, including increased user retention and ongoing engagement as well as rapid adoption [6]. Achievement unlocking feature encouraged users to be active, surprising DLA who thought what took only 6 months would take twice as long. Users called it addictive and fun, using the training content.

Perhaps the strongest support and advancement to gamification was from Salesforce whose add-on, Engage, offered direct competition in an organization by tying user activities with several game mechanics. Organizations who aim to capitalize on surfacing different behaviors may incorporate this level of competitive visibility, and hopefully drive additional engagement with their systems.



Mark Fuller
13842 Points
Level 10
77% to the next level
Recent Achievements
Current Challenges
- Daily Calls
- Advancing Opportunities

Quarterly Stats:
Closed Revenue: $163k
#1
Avg: $124k
Pipeline Revenue: $362k
#1
Avg: $206k
Opportunities Created: 15
#3
Avg: 12
Opportunities Closed: 6
#1

Mark Fuller: 13842 Points

Position

Overall	Quarterly	Monthly	Weekly
#1	#1	#3	#5

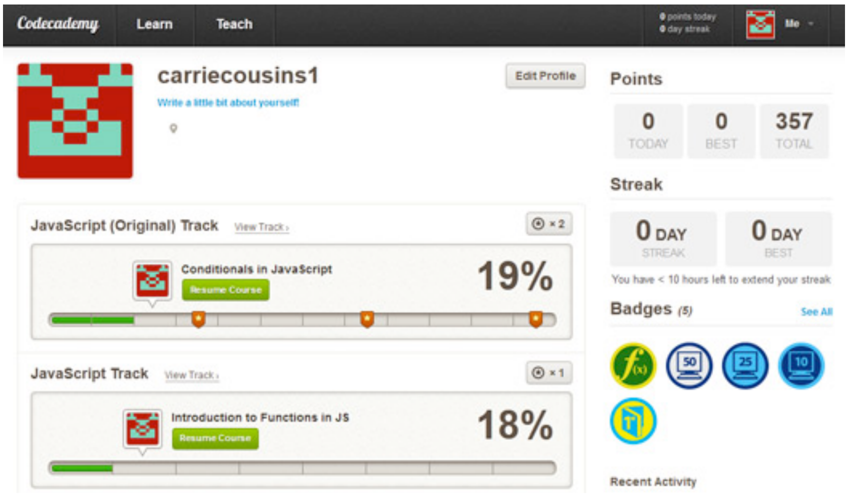
Level 10
77% to the next level
Recent Achievements

Statistics

Challenges Completed	Challenges Available	Achievements Completed	Achievements Remaining
19	2	6	34

Figure 4: Salesforce CRM Gamification[7]

Celebrating elements of their students' progress through a course by using awarding badges for key achievements as well as encouraging them to share these on social networks, several education sites such as Khan Academy and Codecademy compelled their students' to continue their learning [8].



carriecousins1
Write a little bit about yourself!

JavaScript (Original) Track ×2
Conditionals in JavaScript
19%

JavaScript Track ×1
Introduction to Functions in JS
18%

Points

TODAY	BEST	TOTAL
0	0	357

Streak

STREAK	BEST
0 DAY	0 DAY

You have < 10 hours left to extend your streak

Badges (5)

Figure 5: Codecademy Gamification [8]

2.2 Cloud Computing Studies

Cloud computing is composed of different approaches, which are related with the idea of providing everything as a service. Following directions of cloud computing could be examined [9]; Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and finally Software as a Service (SaaS).

[10] Offerings with *as a service* as a suffix include traits like the following:

- Low barriers to entry, making them available to small businesses
- Large scalability
- Multitenancy, which allows resources to be shared by many users
- Device independence, which allows users to access the systems on different hardware [10]

There are several deployment models, concerning how these services might be provided: public, hybrid and private [11]. In the figure 6 the comprehensive model, describing relations between services, uses and types of clouds, is presented [11] .

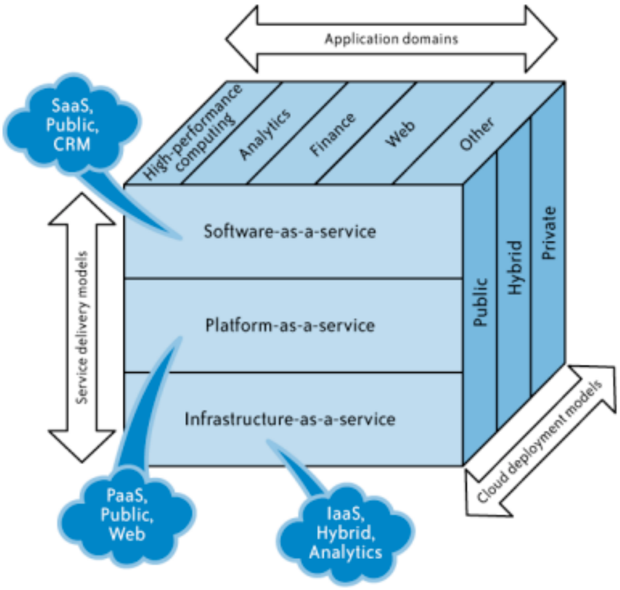


Figure 6: Comprehensive cloud model [11]

[12] The majority of public cloud deployments are generally used for web servers or development systems where security and compliance requirements of larger organizations and their customers is not an issue. [12] Private cloud computing, on the other hand, by definition is a single-tenant environment where the hardware, storage and network are dedicated to a single client or company.

[12] The public cloud is defined as a multi-tenant environment, where you buy a "server slice" in a cloud computing environment that is shared with a number of other clients or tenants.

There is a layered aspect in this categorization, hardware layer, which is composed of network, hardware and operating system resources. Platform layer, which is composed of application server runtimes and database, and last but not least software layer which is the software application that creates the value add for end users.

Figure 7 is depicts the aforementioned layering.

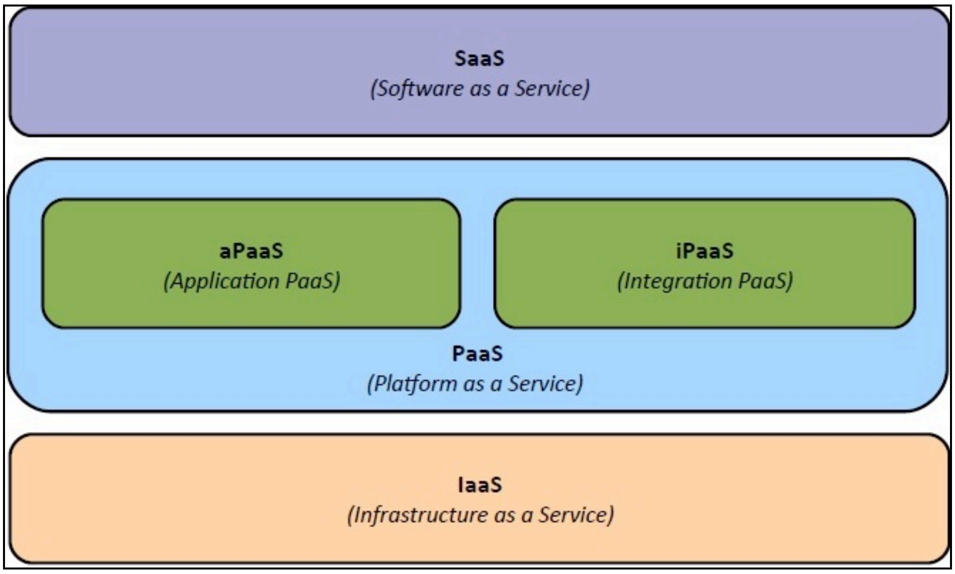


Figure 7: Cloud Computing Services

All the layers above serve to different end user groups. IaaS layer is for system administrators, engineers basically who would like to allocate computing power over the cloud for their specific computing needs.

PaaS layer serves to developers who would like to spare computing platforms for the apps they built using a specific programming language. For instance, a Java developer builds an app using Java, and allocates a Tomcat runtime using a PaaS, like Rackspace. Developer in this scenario is not bothered with underlying infrastructure details, network topologies, machine configuration.

SaaS layer on the other hand, serves to business users, and enables them to use software applications to execute their business processes. For instance, a production company uses stock management software to keep track of their produced items. They analyze the stock and processes at any time for production loss, and make better decisions.

2.2.1 Software as a Service (SaaS)

Software as a Service model enables the end users to access software functionality through web browsers and internet. It is the easiest deployment model when compared to packaged software. There are numerous SaaS offerings available in the market today, but still the cloud software market is expected to grow until year 2020 according to IDC [13]. This proves how promising SaaS model is, and also shows us the fact that there is still time for market saturation. To cope with the increasing demand of SaaS solutions, there should be ways to increase app offerings in the market. For that, there should be ways to engage app developers to develop SaaS apps for corporate needs. This thesis focuses on this, proposes a model to increase developer interaction with SaaS development. From a practitioner perspective, the purpose of this thesis is fully aligned with future growth of the cloud market.

There are few SaaS solutions that have dominated the SaaS market in recent years; Salesforce, Netsuite and Zoho Apps. The advantage of SaaS model is that all applications can be tried before actual use; also they're purchased over a subscription basis, which is mostly monthly. So there is little room for making wrong investment decisions, there is a chance to give up in case of dissatisfaction. SaaS applications also are accessible from any computer on the Internet and enables collaboration among the work force. This is also a unique proposition. In this thesis, we will focus on collaboration over an app development as a service platform, which makes it even more unique.

2.2.2 Platform as a Service (PaaS)

PaaS encompasses the middleware components of an application architecture, mainly application runtimes, databases, queuing services.

PaaS cannot be considered as an independent layer, but coupling with IaaS layer should be underlined. PaaS can only be a value creating service if it is combined with managed infrastructure services. PaaS became an independent service provided to developer community, since developers look for ways of provisioning of application architectures and system topologies the easier way, while sticking to their preferred programing languages and runtime choices.

PaaS offerings are multi-tenant by nature. A developer using a PaaS to build their app topology, will not be intervened by another developers choices of infra services, but stay in total isolation. On the other hand, PaaS provides operational services like security, scalability, metering of resources, high availability, also in some PaaS types it includes service like app development, testing, etc. This kind of PaaS is called aPaaS, and in this thesis we will focus on PaaS of this kind.

PaaS solution providers take the responsibility for managing app runtimes, databases, middleware, security, performance, availability, reliability, and scalability. PaaS makes it easier for the developer to setup the right system topology with the right set of operational features.

Gartner released a report that PaaS was broken into two categories [14]: application Platform as a Service (aPaaS) and Integration Platform as a Service (iPaaS).

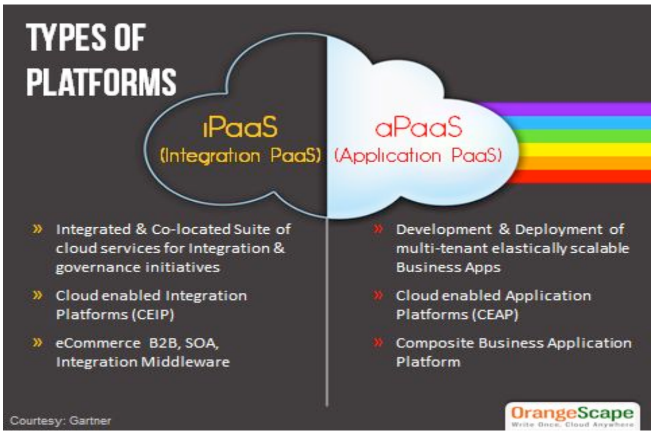

TYPES OF PLATFORMS
iPaaS
(Integration PaaS)
aPaaS
(Application PaaS)
» Integrated & Co-located Suite of cloud services for Integration & governance initiatives
» Cloud enabled Integration Platforms (CEIP)
» eCommerce B2B, SOA, Integration Middleware
» Development & Deployment of multi-tenant elastically scalable Business Apps
» Cloud enabled Application Platforms (CEAP)
» Composite Business Application Platform
Courtesy: Gartner
OrangeScape
Write Once, Cloud Anywhere

Figure 8: PaaS subcategories: iPaaS and aPaaS [15]

2.2.2.1. Integration Platform as a Service (iPaaS)

As more companies move their business into SaaS, there becomes more need for integration among these apps. Either SaaS to local apps, or SaaS to SaaS integration becomes the need. Integration platform as a service (iPaaS) is introduced for those solutions that address SaaS to SaaS integration needs. Since iPaaS is not in the scope of this thesis, this section is not detailed more than this on purpose.

2.2.2.2. Application Platform as a Service (aPaaS)

Below is the picture from Gartner that depicts the categories within the Platform as a Service (PaaS) layer.

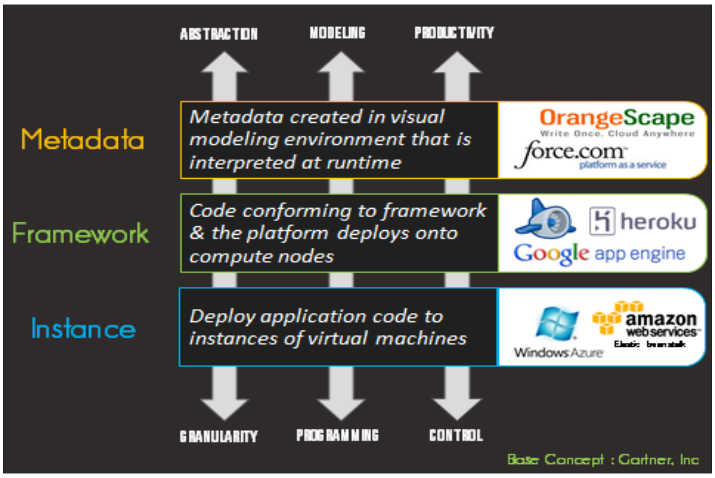

ABSTRACTION
MODELING
PRODUCTIVITY
Metadata
Metadata created in visual modeling environment that is interpreted at runtime
OrangeScape
Write Once, Cloud Anywhere
force.com
platform as a service
Framework
Code conforming to framework & the platform deploys onto compute nodes
heroku
Google app engine
Instance
Deploy application code to instances of virtual machines
Windows Azure
amazon web services
Elastic beanstalk
GRANULARITY
PROGRAMMING
CONTROL
Base Concept : Gartner, Inc

Figure 9: Categories of aPaaS [15]

We shall discuss 3 types of PaaS offerings below.

2.2.2.2.1. Instance PaaS

Instance PaaS is the initial definition of PaaS when it was first introduced. It refers to platform infrastructure services to deploy the application into a cloud environment. In other simple terms, instance PaaS is what you need to "cloudify" your application. Just to name a few PaaS offerings; Amazon Beanstalk, Heroku, Microsoft Azure, Openstack based Rackspace, CloudBees are of this kind.

2.2.2.2.2. Framework PaaS

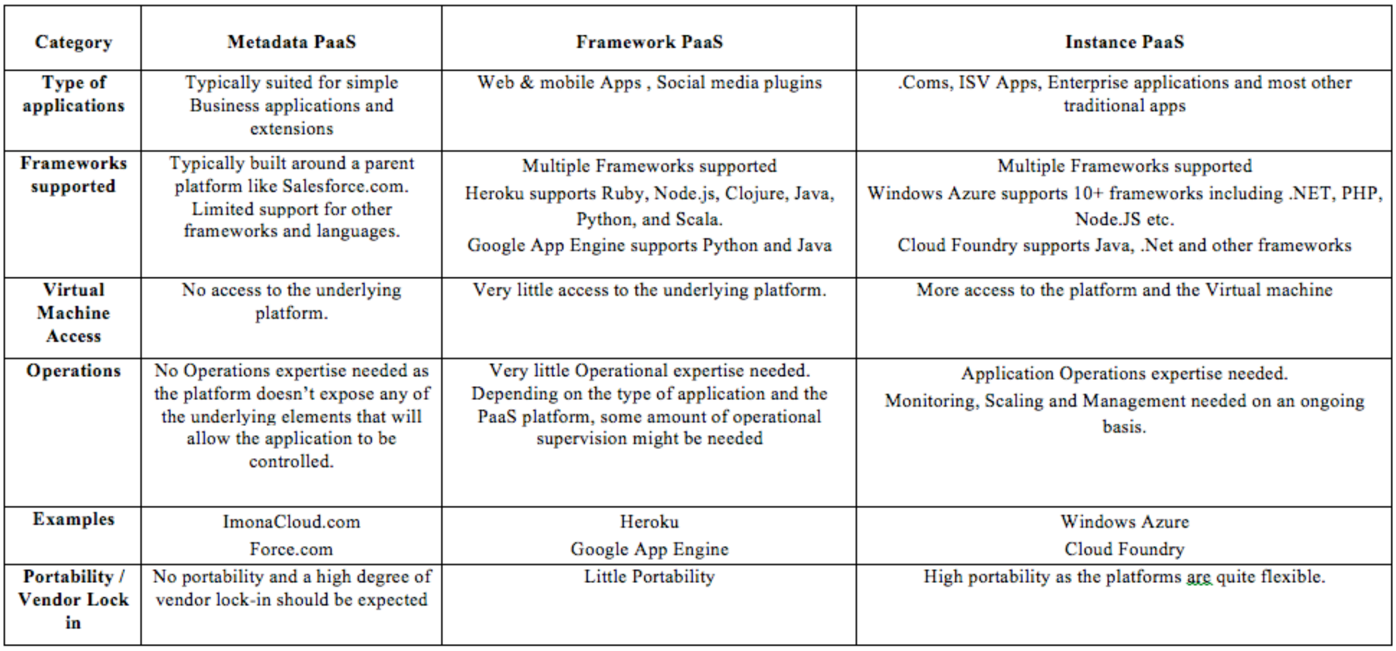
Google has a service called Google App Engine that provides a framework for developers to adhere to, in case they want their apps to make most use of Google Cloud Services. There are limitations in terms the libraries or programming paradigms to use while building an app in Google App engine, however on the other hand it is relatively easier to implement multi-tenant and highly saleable apps in this cloud platform.

2.2.2.2.3. Metadata PaaS

Salesforce is a big player on SaaS layer, however they're also active in Metadata PaaS layer with their force.com product. ImonaCloud (imona.com) is a Turkish company developing a metadata PaaS. Metadata-PaaS provider their developer users with development tools to make it easier and faster to build SaaS solutions. Abstraction layer of app development using a metadata PaaS is even higher than all other cloud layers underneath, however limitations are more too. There are constraints as to how the app looks like, or e.g. to migrate to another UI look & feel, it should be the platform's decision first to migrate, rather than the individual developer. On the contrary this should be an advantage in terms of decreasing the migration cost to a different look & feel, functionality or usability. When the metadata PaaS vendor makes a change in their platform and introduces a feature, all the apps built using this PaaS inherit the feature immediately with no cost incurred.

All in all, the advantage of a metadata PaaS is that all applications can be developed rapidly at a low cost. On the other hand, metadata PaaS limits developers to a certain programming languages and tools. However if you make a choice to build your application using Phyton, you lock in your solution to Phyton, likewise if you use a certain metadata PaaS to build your solution, you build it faster than other traditional 3GL programming languages, but you lock your solution into the platform capabilities.

Metadata PaaS provides visual tools to design and model apps and store their metadata. Also it provides an application runtime that interprets the metadata and executes the application using the best usability for the end user. Metadata PaaS provides more control over developer activities, and can capture any developer movement over the platform during the development phase. Tha t is why in this thesis, we will focus on a Metadata PaaS, because by the help of tracking developer movements, we would be able to devise an engagement model using gamification techniques.



Category	Metadata PaaS	Framework PaaS	Instance PaaS
Type of applications	Typically suited for simple Business applications and extensions	Web & mobile Apps , Social media plugins	.Coms, ISV Apps, Enterprise applications and most other traditional apps
Frameworks supported	Typically built around a parent platform like Salesforce.com. Limited support for other frameworks and languages.	Multiple Frameworks supported Heroku supports Ruby, Node.js, Clojure, Java, Python, and Scala. Google App Engine supports Python and Java	Multiple Frameworks supported Windows Azure supports 10+ frameworks including .NET, PHP, Node.JS etc. Cloud Foundry supports Java, .Net and other frameworks
Virtual Machine Access	No access to the underlying platform.	Very little access to the underlying platform.	More access to the platform and the Virtual machine
Operations	No Operations expertise needed as the platform doesn't expose any of the underlying elements that will allow the application to be controlled.	Very little Operational expertise needed. Depending on the type of application and the PaaS platform, some amount of operational supervision might be needed	Application Operations expertise needed. Monitoring, Scaling and Management needed on an ongoing basis.
Examples	ImonaCloud.com Force.com	Heroku Google App Engine	Windows Azure Cloud Foundry
Portability / Vendor Lock in	No portability and a high degree of vendor lock-in should be expected	Little Portability	High portability as the platforms are quite flexible.

Figure 10: Comparison of PaaS solutions [17]

Chapter 3

Model Proposed

3.1 MDA

One of the most frequently leveraged frameworks of game design is referred to as MDA—which stands for [1]:

- Mechanics
- Dynamics
- Aesthetics

Mechanics describe the base components of the game. At their core, they allow a designer control over the levels of the game, giving her the ability to guide player actions [1].

Dynamics describe player's behavior with these mechanics. They determine what each player is doing in response to the mechanics of the system, both individually and with other players. Occasionally, game mechanics and game dynamics are used interchangeably, but they are considerably different [1].

Finally, the ***aesthetics*** describe as how the game makes the player feel when he/she interacts with the game system. Game aesthetics can be viewed as the composite outcome of the mechanics and dynamics as they interact with and create emotions [1].

After examining the ImonaCloud platform, we came up with the mechanics of the platform. After that we defined readable actions for those developers that practice these mechanics. This is the dynamics part of the gamification experience.

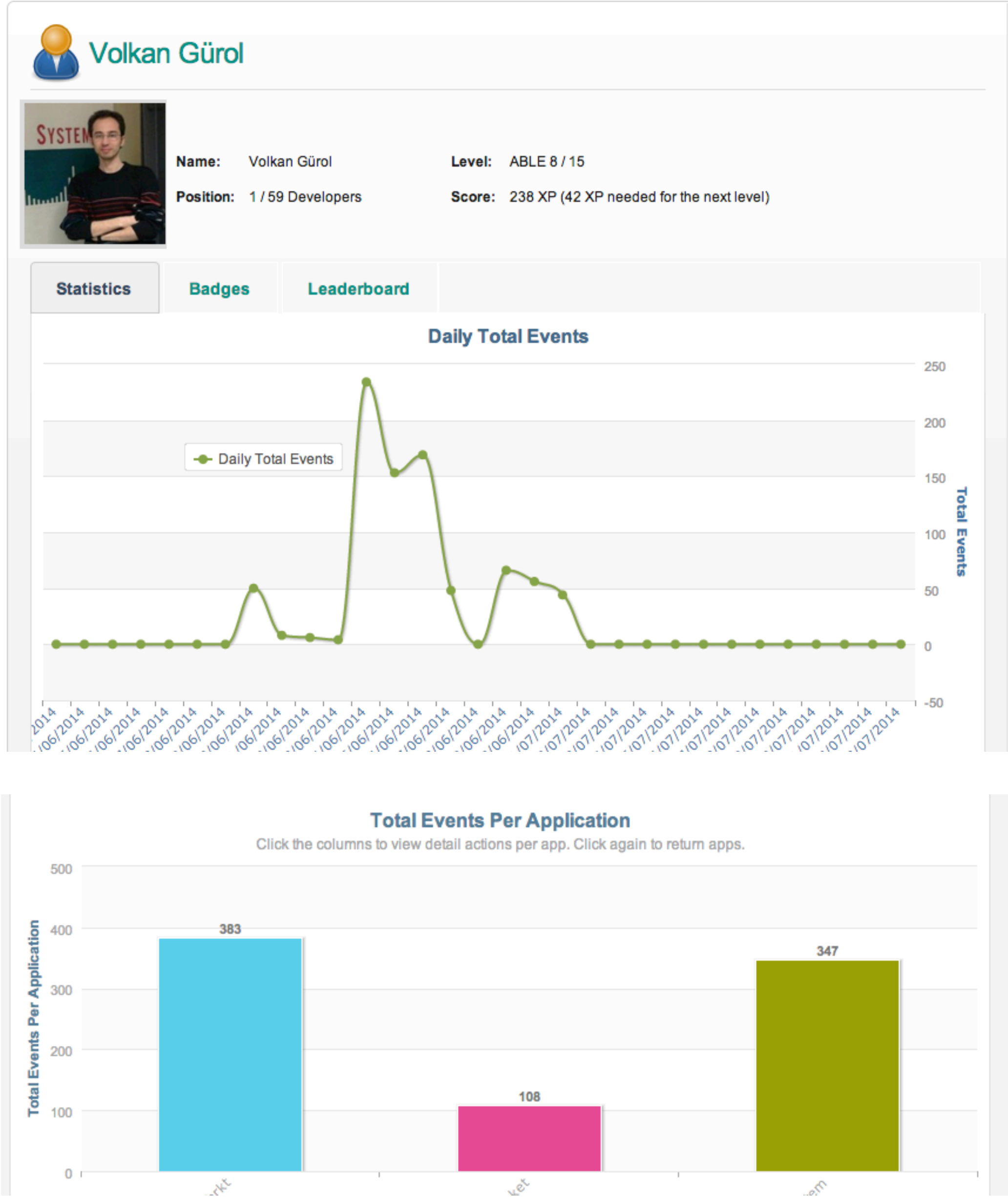
3.2 ImonaCloud Gamification Model

We integrated 4 gamification elements on ImonaCloud platform; Points, Badges, Levels and Leaderboard system. We contend that both platform usage performance and marketplace performance effect developer's score and reputation. Platform usage performance consists of learning platform capabilities and efficiency of actions on the platform. Significant meta-data actions were categorized according to their difficulty levels and designed as badges. Earning badges make developers engage more with the platform. In that way, developers can learn platform capabilities faster during the app development process.

Marketplace performance also plays a critical role in scoring. Weekly total app sold and application reviews received from customers contribute to the overall developer score.

- **Platform Usage Performance**
    - Reward with points for the actions developer makes according to actions' difficulty level
    - Give badges by completing specific challenges
    - Show log of metadata actions developer makes
- **Marketplace Performance**
    - Give experience points for weekly total apps sold
    - Give points for application reviews received (positive/negative reviews affect points)

We created a profile page for developers, so that they can track their activities, levels, scores and badges easily. Developers can keep track of their daily performances on the platform and compare them.



Volkan Gürol

Name: Volkan Gürol

Position: 1 / 59 Developers

Level: ABLE 8 / 15

Score: 238 XP (42 XP needed for the next level)

Statistics
Badges
Leaderboard
Daily Total Events
Daily Total Events
Total Events
250
200
150
100
50
0
-50
Total Events Per Application
Click the columns to view detail actions per app. Click again to return apps.
Total Events Per Application
500
400
300
200
100
0
383
108
347

Figure 11: Developer’s profile page

As is seen in Figure 11, developer gained 238 points and the leader among 59 developers. Developers can also see each event they made on per application.

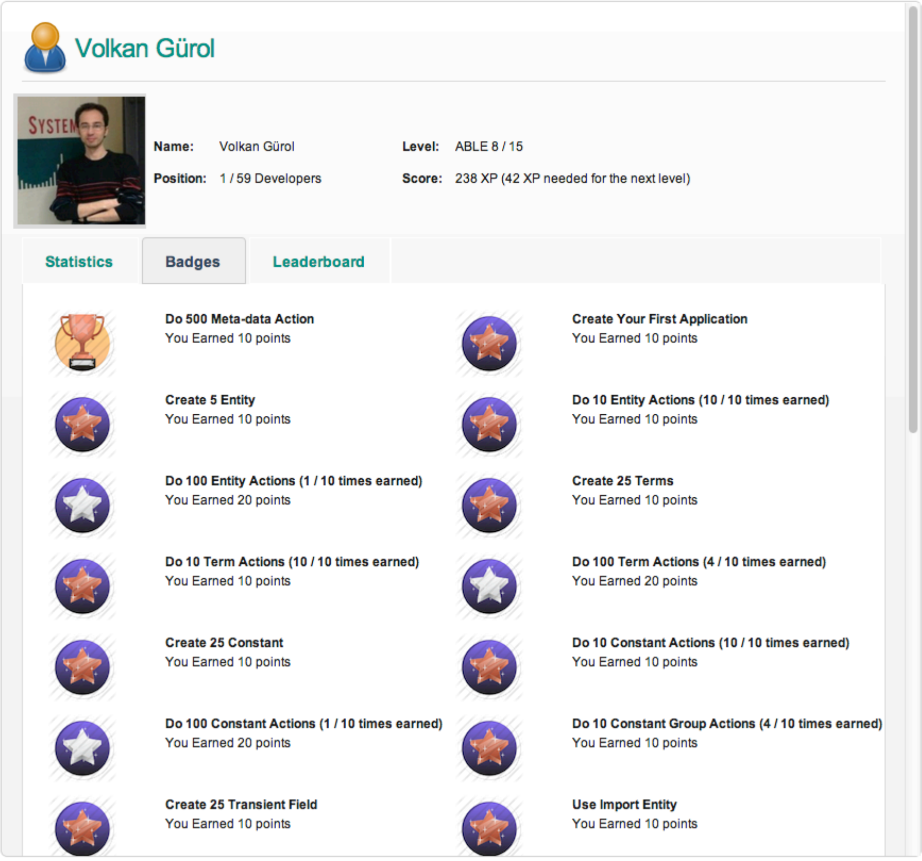
3.2.1 Points

Points are rewarded to developers for their efficient and valuable activities on the platform. Giving points and badges on every platform actions as gamification dynamics, although might sound like entertaining and being responsive as a platform, have a fairly bad reputation in recent years. One of the reasons is that they are overused. In our platform, we tried to balance the point system and trophies properly. Developers can earn points by doing platform meta-data actions and marketplace actions.

In the scope of this thesis, we followed two-step in our gamification model. In the first phase, we worked with 5 people that 3 of them had no on programming, the other 2 of them had experience on app development before. By tracking their meta-data actions, we defined valuable actions on that platform in order to distribute points fairly. In the second phase, we tested our proposed scoring model on a large group that 11 teams consisted of 2-3 people in Hackathon Contest, which was held in ITU Çekirdek. We analyzed each developer's performance whether the proposed model was viable for that kind of aPaaS platforms. Our point system will be described in Data Analysis, Findings and Discussion chapter.

3.2.2 Badges

Badges refer to achievement artifacts in the platform. Significant development milestones are designed as badges. When developer earns an achievement as a badge, they recognize the fact that they raised the challenge bar up one level. They would be proud of holding that badge, and look for earning more.



Volkan Gürol

Name: Volkan Gürol

Position: 1 / 59 Developers

Level: ABLE 8 / 15

Score: 238 XP (42 XP needed for the next level)

Do 500 Meta-data Action
You Earned 10 points

Create Your First Application
You Earned 10 points

Create 5 Entity
You Earned 10 points

Do 10 Entity Actions (10 / 10 times earned)
You Earned 10 points

Do 100 Entity Actions (1 / 10 times earned)
You Earned 20 points

Create 25 Terms
You Earned 10 points

Do 10 Term Actions (10 / 10 times earned)
You Earned 10 points

Do 100 Term Actions (4 / 10 times earned)
You Earned 20 points

Create 25 Constant
You Earned 10 points

Do 10 Constant Actions (10 / 10 times earned)
You Earned 10 points

Do 100 Constant Actions (1 / 10 times earned)
You Earned 20 points

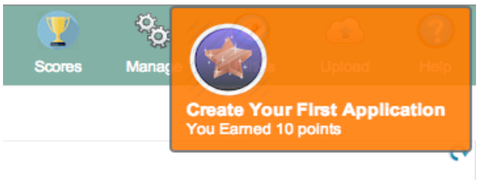
Do 10 Constant Group Actions (4 / 10 times earned)
You Earned 10 points

Create 25 Transient Field
You Earned 10 points

Use Import Entity
You Earned 10 points

Figure 12: Earning Badges shows in the profile

Each earned badge displayed with notification icon and rewards with points. We categorized achievements in terms of their difficulty level; bronze, silver and gold badges.

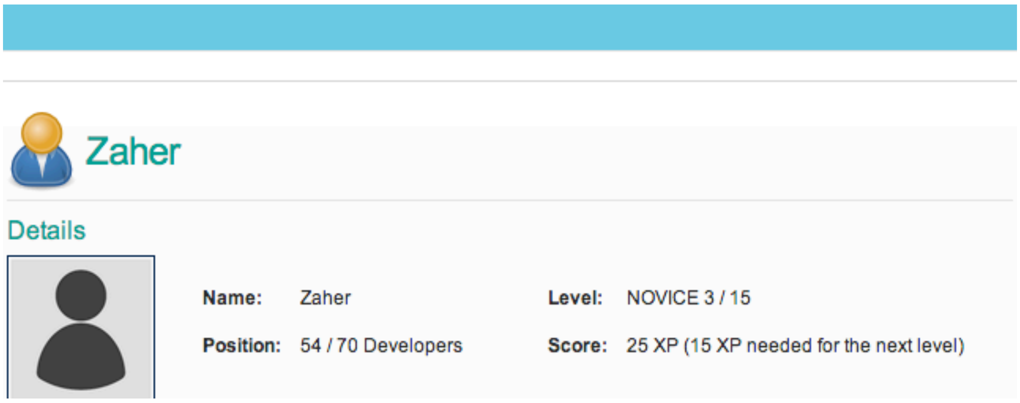


Create Your First Application
You Earned 10 points

Figure 13: When a developer earns a badge, it will show up as notification.

3.2.3 Levels

Levels are a system, by which developers are rewarded an increasing value for a cumulating of points. Leveling is one of the highest components of motivation for developers. In ImonaCloud platform we showed developer's level in profile detail. Also developers can see how many required XP remains for the next level.



Zaher

Details

Name: Zaher

Level: NOVICE 3 / 15

Position: 54 / 70 Developers

Score: 25 XP (15 XP needed for the next level)

Figure 14: Level System.

In platform, each of level that is designed as exponentially growth, describe developer's proficiencies. Developers will have to spend more time for reaching higher levels. There are currently 15 levels defined in the platform.

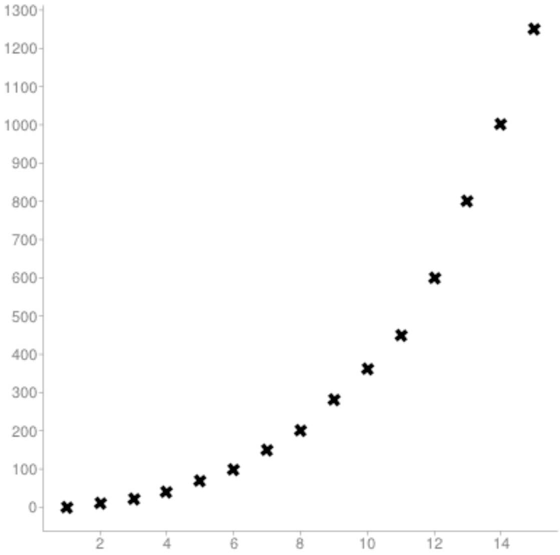


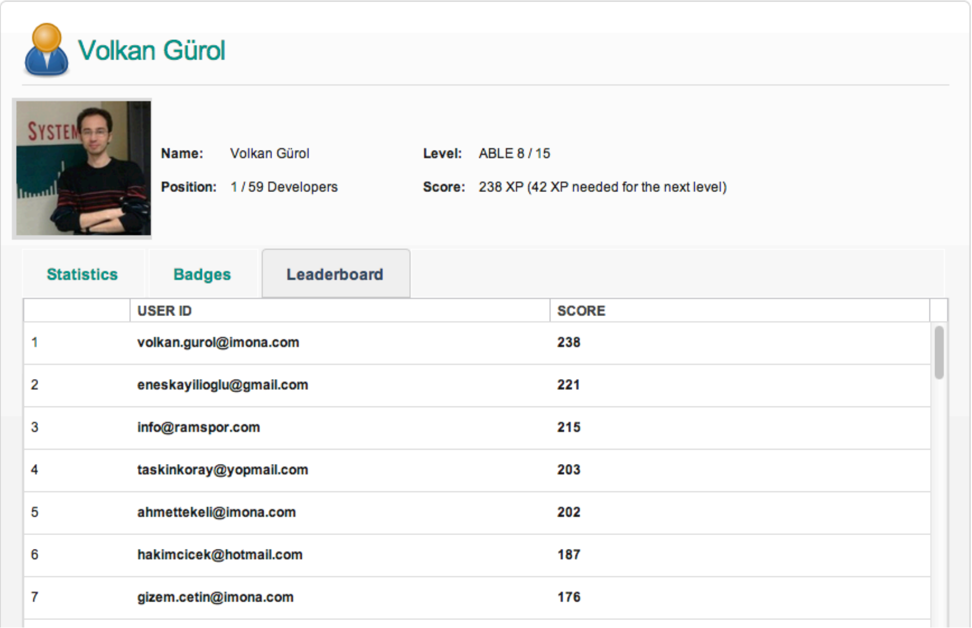
Figure 15: Required Level Points

Levels	Level Name	Range Points	
Level 1	NEWBIE	0	9
Level 2	BEGINNER	10	19
Level 3	NOVICE	20	39
Level 4	AMATEUR	40	69
Level 5	APPRENTICE	70	99
Level 6	QUALIFIED	100	149
Level 7	TRAINED	150	199
Level 8	ABLE	200	279
Level 9	COMPETENT	280	359
Level 10	CAPABLE	360	449
Level 11	SKILLED	450	599
Level 12	EXPERIENCED	600	799
Level 13	PROFICIENT	800	999
Level 14	VETERAN	1000	1249
Level 15	CHAMPION	1250	1500

Table 1: Level Points System

3.2.4 Leaderboards

Leaderboards shows where a developer stands relative to other developers. We designed a leaderboard called “Top Developers Leaderboard”, which also shows the current developers ranking amongst the others. Top developers leaderboard shows the most success developers on the platform. This leaderboard may be used in the market, so that employers can find the right developers for the right job.



Volkan Gürol

Name: Volkan Gürol

Level: ABLE 8 / 15

Position: 1 / 59 Developers

Score: 238 XP (42 XP needed for the next level)

	USER ID	SCORE
1	volkan.gurol@imona.com	238
2	eneskayilioglu@gmail.com	221
3	info@ramspor.com	215
4	taskinkoray@yopmail.com	203
5	ahmettekeli@imona.com	202
6	hakimcicek@hotmail.com	187
7	gizem.cetin@imona.com	176

Figure 16: Top Developers Leaderboard

3.2.5 Stats

Every developer who is in learning process of a new paradigm would like to know his or her progress and get feedback. With the progress charts, they will be able to track their performance in daily basis. When a developer monitors their progress, we believe that they will reach to a higher motivation.

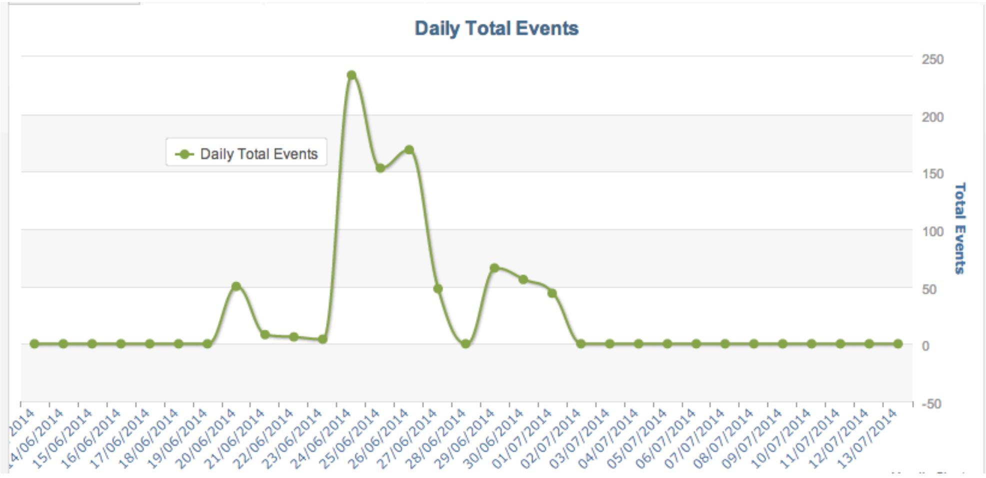
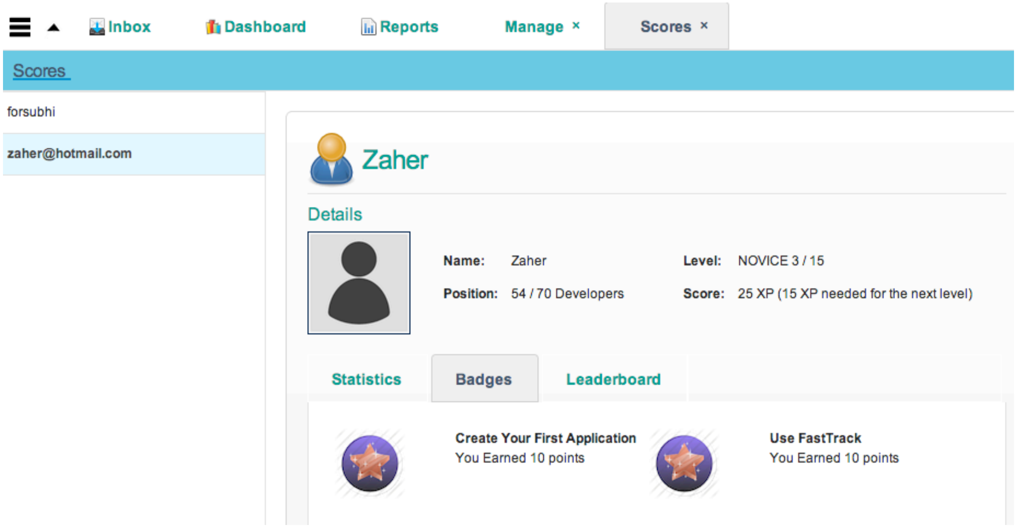


Figure 17: Daily developer's performance history

Also, in case there is more than one developer in the same organization, there is a chance to compare performance amongst them. Developers will show in the left side menu.



forsubhi

zaher@hotmail.com

Zaher

Details

Name: Zaher

Position: 54 / 70 Developers

Level: NOVICE 3 / 15

Score: 25 XP (15 XP needed for the next level)

Create Your First Application
You Earned 10 points

Use FastTrack
You Earned 10 points

Figure 18: Compare per developer's productivity

Developed apps are shown in statistics page and by clicking on a specific app, developers can see the all metadata actions they created in his app during engagement with the platform.

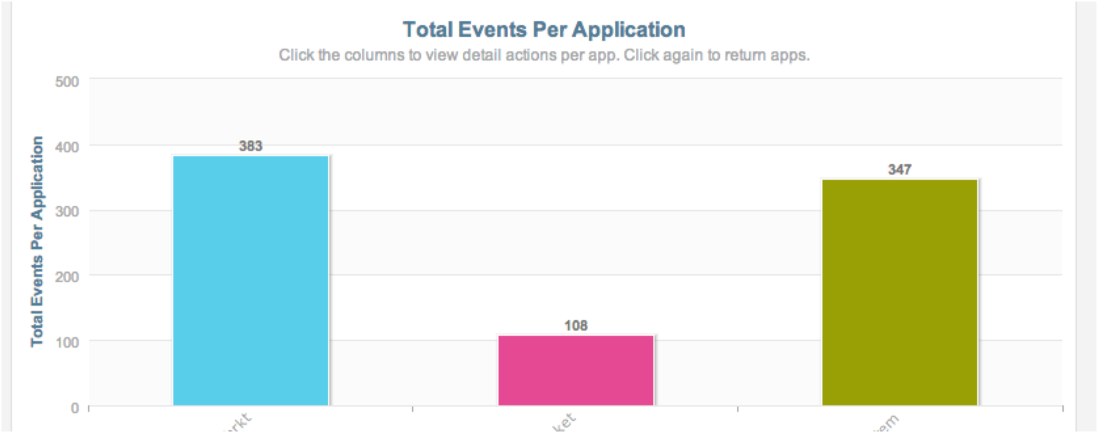


Figure 19: All developed apps

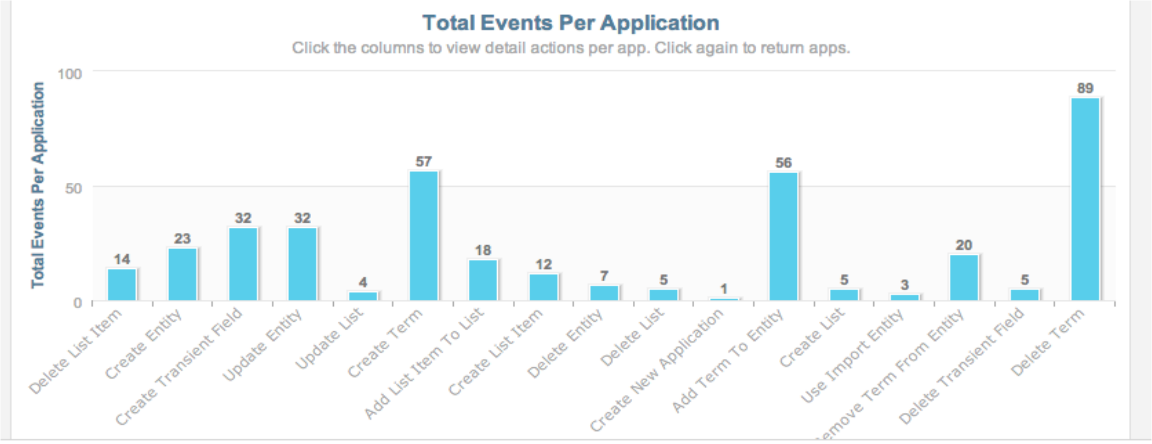


Figure 20: All developed metadata actions related to app

We consider that these gamification techniques will help developers to learn the platform faster and make them engage with it.

3.3 Database Design

To comprehend the fundamentals of scoring a developer's relevant actions on a metadata PaaS, it makes sense to delve into the core design principles of the metadata's database model of the platform. Below is the table of the entities we designed for the scope of this thesis to model the solution from a database modeling perspective. Note that some of the tables introduced below are a sneak preview of the core tables that already belong to the platform that defines metadata, and others related to the scoring are introduced in the scope of this thesis.

Table Names	Description
imona_business_app	Stores apps belong to organization.
imona_business_app_entities	Stores entities corresponding to apps.
imona_entity	Stores all entities.
imona_entity_terms	Stores terms belong to entities.
imona_base_token	Stores all terms', lists', menus' etc. details.
imona_dev_events	All Meta-data actions are stored

Table 2: DB Table List

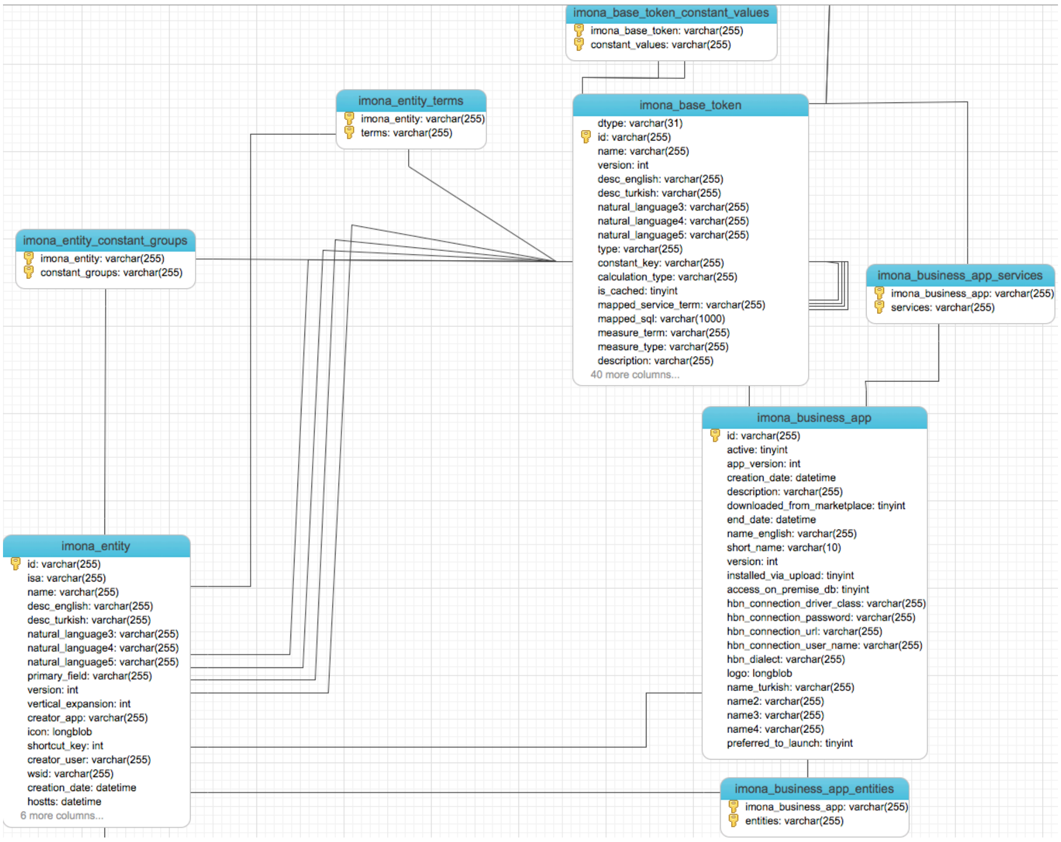
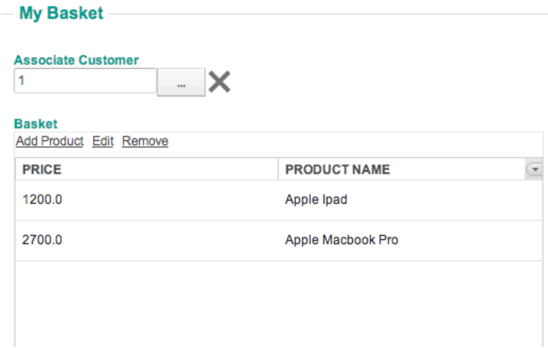

imona_base_token_constant_values
imona_base_token: varchar(255)
constant_values: varchar(255)
imona_entity_terms
imona_entity: varchar(255)
terms: varchar(255)
imona_base_token
dtype: varchar(31)
id: varchar(255)
name: varchar(255)
version: int
desc_english: varchar(255)
desc_turkish: varchar(255)
natural_language3: varchar(255)
natural_language4: varchar(255)
natural_language5: varchar(255)
type: varchar(255)
constant_key: varchar(255)
calculation_type: varchar(255)
is_cached: tinyint
mapped_service_term: varchar(255)
mapped_sql: varchar(1000)
measure_term: varchar(255)
measure_type: varchar(255)
description: varchar(255)
40 more columns...
imona_entity_constant_groups
imona_entity: varchar(255)
constant_groups: varchar(255)
imona_business_app_services
imona_business_app: varchar(255)
services: varchar(255)
imona_business_app
id: varchar(255)
active: tinyint
app_version: int
creation_date: datetime
description: varchar(255)
downloaded_from_marketplace: tinyint
end_date: datetime
name_english: varchar(255)
short_name: varchar(10)
version: int
installed_via_upload: tinyint
access_on_premise_db: tinyint
hbn_connection_driver_class: varchar(255)
hbn_connection_password: varchar(255)
hbn_connection_url: varchar(255)
hbn_connection_user_name: varchar(255)
hbn_dialect: varchar(255)
logo: longblob
name_turkish: varchar(255)
name2: varchar(255)
name3: varchar(255)
name4: varchar(255)
preferred_to_launch: tinyint
imona_entity
id: varchar(255)
isa: varchar(255)
name: varchar(255)
desc_english: varchar(255)
desc_turkish: varchar(255)
natural_language3: varchar(255)
natural_language4: varchar(255)
natural_language5: varchar(255)
primary_field: varchar(255)
version: int
vertical_expansion: int
creator_app: varchar(255)
icon: longblob
shortcut_key: int
creator_user: varchar(255)
wsid: varchar(255)
creation_date: datetime
hostts: datetime
6 more columns...
imona_business_app_entities
imona_business_app: varchar(255)
entities: varchar(255)

Figure 21: Database Design

To illustrate how data is stored in the platform's metadata database, we will explain using an example by building a simple application. In this simple business scenario, there are customers, products and customers are adding products into their baskets.

We created three main entities: "Customer", "Product" and "My Basket". Customer entity stores the buyers; Product entity stores the product items and their price; My Basket entity stores the data that which customers added product/products in their baskets. In that example, we associated customer with 1-1 relation and created basket list as M-M relation.



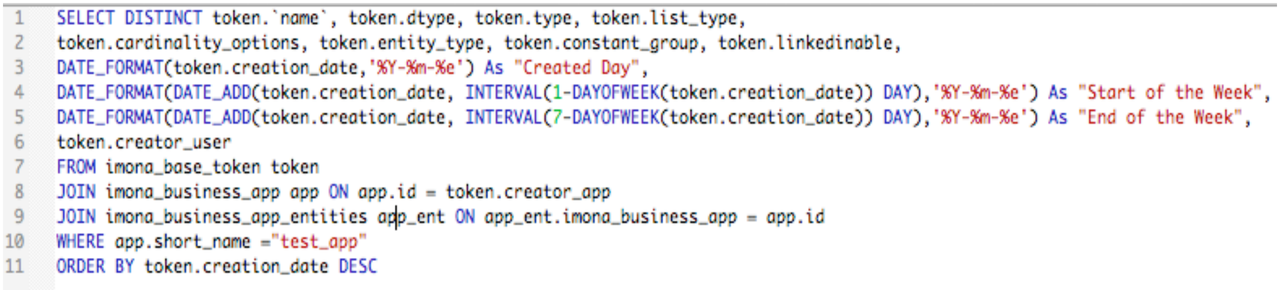
Associate Customer
1

Basket

PRICE	PRODUCT NAME
1200.0	Apple Ipad
2700.0	Apple Macbook Pro

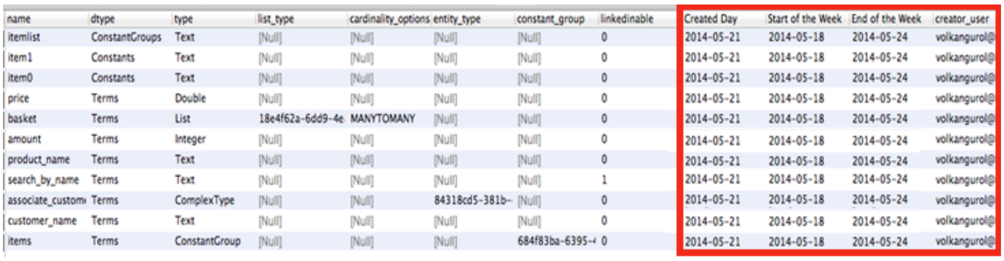
Figure 22: My Basket Entity Form

We listed created terms by writing a query.



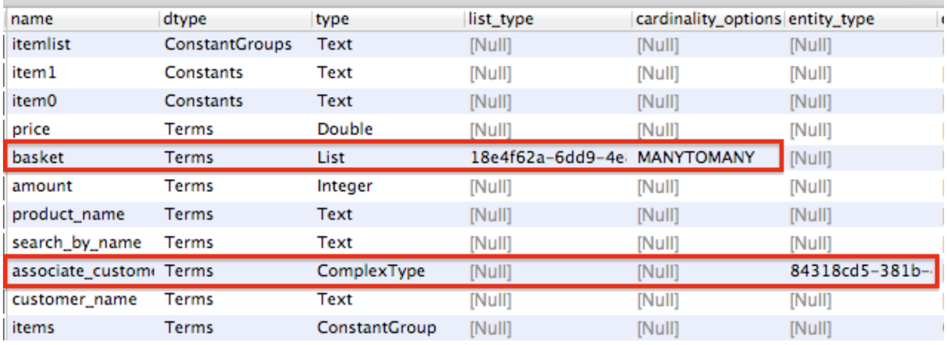
```sql
SELECT DISTINCT token.`name`, token.dtype, token.type, token.list_type,
token.cardinality_options, token.entity_type, token.constant_group, token.linkedinable,
DATE_FORMAT(token.creation_date,'%Y-%m-%e') As "Created Day",
DATE_FORMAT(DATE_ADD(token.creation_date, INTERVAL(1-DAYOFWEEK(token.creation_date)) DAY),'%Y-%m-%e') As "Start of the Week",
DATE_FORMAT(DATE_ADD(token.creation_date, INTERVAL(7-DAYOFWEEK(token.creation_date)) DAY),'%Y-%m-%e') As "End of the Week",
token.creator_user
FROM imona_base_token token
JOIN imona_business_app app ON app.id = token.creator_app
JOIN imona_business_app_entities app_ent ON app_ent.imona_business_app = app.id
WHERE app.short_name ="test_app"
ORDER BY token.creation_date DESC
```

Figure 23: Sql Query for listing meta-data



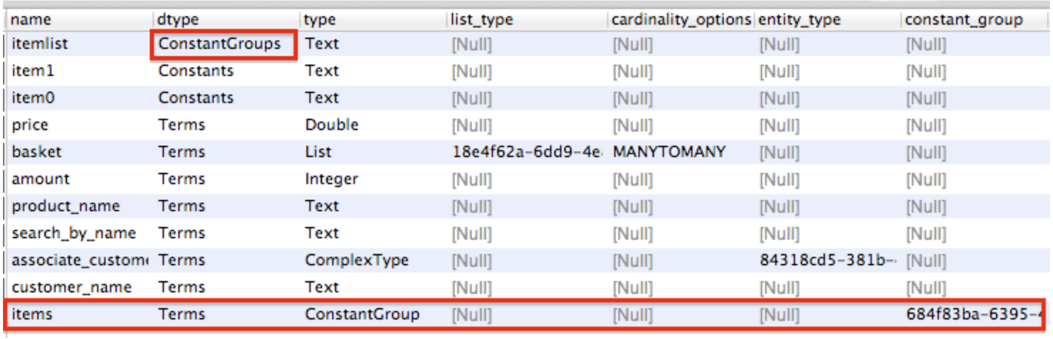
name	dtype	type	list_type	cardinality_options	entity_type	constant_group	linkedinable	Created Day	Start of the Week	End of the Week	creator_user
itemlist	ConstantGroups	Text	[Null]	[Null]	[Null]	[Null]	0	2014-05-21	2014-05-18	2014-05-24	volkangurol@
item1	Constants	Text	[Null]	[Null]	[Null]	[Null]	0	2014-05-21	2014-05-18	2014-05-24	volkangurol@
item0	Constants	Text	[Null]	[Null]	[Null]	[Null]	0	2014-05-21	2014-05-18	2014-05-24	volkangurol@
price	Terms	Double	[Null]	[Null]	[Null]	[Null]	0	2014-05-21	2014-05-18	2014-05-24	volkangurol@
basket	Terms	List	18e4f62a-6dd9-4e	MANYTOMANY	[Null]	[Null]	0	2014-05-21	2014-05-18	2014-05-24	volkangurol@
amount	Terms	Integer	[Null]	[Null]	[Null]	[Null]	0	2014-05-21	2014-05-18	2014-05-24	volkangurol@
product_name	Terms	Text	[Null]	[Null]	[Null]	[Null]	0	2014-05-21	2014-05-18	2014-05-24	volkangurol@
search_by_name	Terms	Text	[Null]	[Null]	[Null]	[Null]	1	2014-05-21	2014-05-18	2014-05-24	volkangurol@
associate_custom	Terms	ComplexType	[Null]	[Null]	84318cd5-381b-	[Null]	0	2014-05-21	2014-05-18	2014-05-24	volkangurol@
customer_name	Terms	Text	[Null]	[Null]	[Null]	[Null]	0	2014-05-21	2014-05-18	2014-05-24	volkangurol@
items	Terms	ConstantGroup	[Null]	[Null]	[Null]	684f83ba-6395-	0	2014-05-21	2014-05-18	2014-05-24	volkangurol@

Figure 24: We can track the term when was created



name	dtype	type	list_type	cardinality_options	entity_type
itemlist	ConstantGroups	Text	[Null]	[Null]	[Null]
item1	Constants	Text	[Null]	[Null]	[Null]
item0	Constants	Text	[Null]	[Null]	[Null]
price	Terms	Double	[Null]	[Null]	[Null]
basket	Terms	List	18e4f62a-6dd9-4e	MANYTOMANY	[Null]
amount	Terms	Integer	[Null]	[Null]	[Null]
product_name	Terms	Text	[Null]	[Null]	[Null]
search_by_name	Terms	Text	[Null]	[Null]	[Null]
associate_custom	Terms	ComplexType	[Null]	[Null]	84318cd5-381b-
customer_name	Terms	Text	[Null]	[Null]	[Null]
items	Terms	ConstantGroup	[Null]	[Null]	[Null]

Figure 25: Many-to-Many and 1-1 (Complex Type) relations are found in the model



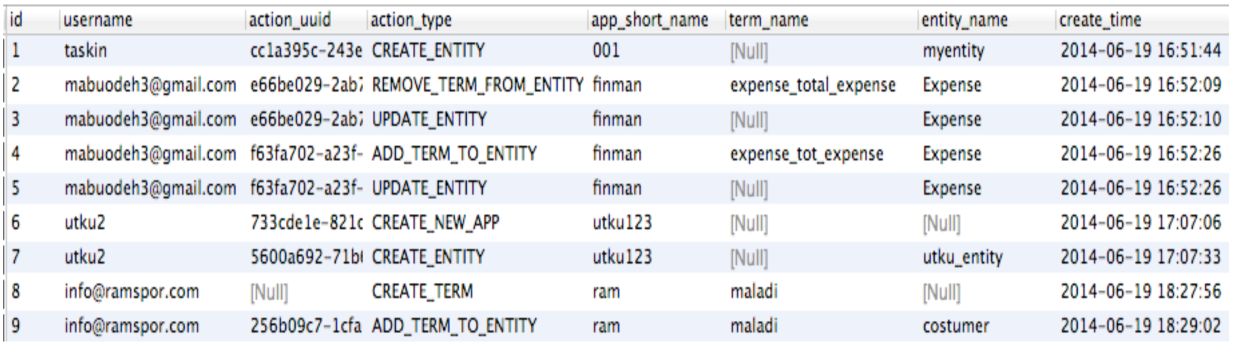
name	dtype	type	list_type	cardinality_options	entity_type	constant_group
itemlist	ConstantGroups	Text	[Null]	[Null]	[Null]	[Null]
item1	Constants	Text	[Null]	[Null]	[Null]	[Null]
item0	Constants	Text	[Null]	[Null]	[Null]	[Null]
price	Terms	Double	[Null]	[Null]	[Null]	[Null]
basket	Terms	List	18e4f62a-6dd9-4e	MANYTOMANY	[Null]	[Null]
amount	Terms	Integer	[Null]	[Null]	[Null]	[Null]
product_name	Terms	Text	[Null]	[Null]	[Null]	[Null]
search_by_name	Terms	Text	[Null]	[Null]	[Null]	[Null]
associate_custom	Terms	ComplexType	[Null]	[Null]	84318cd5-381b-	[Null]
customer_name	Terms	Text	[Null]	[Null]	[Null]	[Null]
items	Terms	ConstantGroup	[Null]	[Null]	[Null]	684f83ba-6395-

Figure 26: This is how items, itemlist relations stored in db.

In order to track developers' all activities including update and delete operations, we created a table "imona_dev_event" that holds almost every actions. To give an example, if developer adds a term to entity we need to know:

- Who did the action?
- At what time action was created?
- On which app action was made?
- Which term was added to entity?
- Did multiple actions at one time or only one action? (Binded with uuid)

In order to clarify this, we basically keep username, action type, creation time, app name and other related things to action like term name, term type, entity name, constant name and so on in "Developer Events Table".



id	username	action_uuid	action_type	app_short_name	term_name	entity_name	create_time
1	taskin	cc1a395c-243e	CREATE_ENTITY	001	[Null]	myentity	2014-06-19 16:51:44
2	mabuodeh3@gmail.com	e66be029-2ab7	REMOVE_TERM_FROM_ENTITY	finman	expense_total_expense	Expense	2014-06-19 16:52:09
3	mabuodeh3@gmail.com	e66be029-2ab7	UPDATE_ENTITY	finman	[Null]	Expense	2014-06-19 16:52:10
4	mabuodeh3@gmail.com	f63fa702-a23f-	ADD_TERM_TO_ENTITY	finman	expense_tot_expense	Expense	2014-06-19 16:52:26
5	mabuodeh3@gmail.com	f63fa702-a23f-	UPDATE_ENTITY	finman	[Null]	Expense	2014-06-19 16:52:26
6	utku2	733cde1e-821c	CREATE_NEW_APP	utku123	[Null]	[Null]	2014-06-19 17:07:06
7	utku2	5600a692-71b	CREATE_ENTITY	utku123	[Null]	utku_entity	2014-06-19 17:07:33
8	info@ramspor.com	[Null]	CREATE_TERM	ram	maladi	[Null]	2014-06-19 18:27:56
9	info@ramspor.com	256b09c7-1cfa	ADD_TERM_TO_ENTITY	ram	maladi	costumer	2014-06-19 18:29:02

Figure 27: This is how developer actions stored in db.

As it is seen above in the figure, we simplified action types so that all actions related to developers hold in one table. The following developer activities are hold in the database:

CRUD ACTION TYPES			
Create New App	Install App	Uninstall App	
Create Entity	Create Transient Entity	Update Entity	Delete Entity
Create Term	Delete Term	Add Term To Entity	Remove Term From Entity
Create Constant	Delete Constant	Add Constant To Constant Group	Remove Constant From Constant Group
Create Constant Group	Update Constant Group	Delete Constant Group	
Create WS Service	Delete WS Service		
Create Rest Service	Delete Rest Service		
Create Service	Delete Service		
Create Report Template	Delete Report Template		
Create Scheduled Jobs	Delete Scheduled Jobs		
Create Transient Field	Delete Transient Field		
USABLE ACTION TYPES			
Use Fasttrack	Use Inheritance	Use Import Entity	Use Import Constant

Table 3: Developer activities

We currently hold 34 valuable meta-data action types now. But it is easily extendable for the future new activities.

Chapter 4

Data Analysis, Findings and Discussion

4.1 Pilot Group 1

In the scope of this thesis, we worked with 3 trainees and 2 app developers in order to define valuable challenges for the developers and distribute points fairly. App developers have knowledge about programming and trainees have no experience on programming before. Firstly, we gave a crash course on how to develop apps on ImonaCloud, and then wanted them to take the online course on udemy.com/imonacloud. Both groups worked on the platform each day a week.

We have observed that,

- While trainees built their first app, they created 250-300 metadata actions. In their second attempt to build an app, they reach up to 500-600 metadata actions; also new UI components were used. However, irrespective of the score they made, their apps were considered to be dissatisfactory by the platform expert, and not nominated to the marketplace
- App developers built 2 integrated apps that each app can be used standalone or both. Dependent app had 1200 terms that mostly contains list, list items (i.e. country-city) and transient objects. Second app was created with 800-900 meta-data actions. Their apps were considered to be satisfactory by the platform expert and now they are in the marketplace.

Main score difference between the groups stemmed from the fact that;

1. Developer created more terms than the trainees
2. Trainees could not manage to execute the actions specified in gold category

We observed that, although trainees did not have experience on programming before, they succeeded building two applications. They used new components in their second application and this shows that they tried new things, features in their application. Our first goal was to engage developers so we will also consider non-developers when defining challenges.

On the other hand, app developers managed to come up with more complex applications. They almost experienced using all field types in the platform. Also they used inheritance in their applications, since they immediately recognized this feature using their knowledge on Object Oriented approach. And as a result their applications managed to get in to the marketplace. Our second goal was to increase the number of apps in the marketplace so that selling apps will be a critical role in our point system.

By looking trainees' activities, we designed 62 valuable platform meta-data action challenges that will help developers for their development process. We separated them in 2 categories like basic actions and CRUD Actions. Also each of these categories has consisted of 3 challenging parts: bronze, silver and gold. Completing a bronze challenge adds 10 XP points to developer's score. Silver challenges are worth 20 XP and completing gold challenges give extra 30 XP.

Earning Type	**Requirements**	**Calculations**	**Points**
Basic - Platform Meta-Data Actions	Creating apps, fields, entities, services…	10 bronze x 10XP 8 silver x 20XP 8 gold x 30XP	Total: 500 XP
CRUD - Platform Meta-Data Actions	Dependent to complexity CRUD meta-data actions, developer earns between 1-5 XP in each challenge.	CRUD actions - 280 XP 12 bronze x 10XP 12 silver x 20XP 12 gold x 30XP	Total: 1000 XP
Marketplace Actions	Earn points as your app is purchased	Dependent to selling apps	Weekly Calculated (10XP-50XP Bonus)

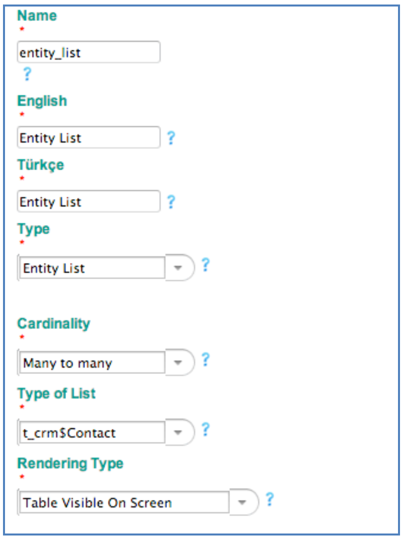
Table 4: Earning Point Types

While, at most 1500 XP can be earned from the platform metadata-actions, marketplace actions have no limitations on gaining XP. As it is understood, selling apps performance plays a critical role in the overall score. We will normalize the weekly total purchased apps related to total customers so that points will be given fairly. To give an example assumes that there are 100 registered potential customers in the market. Each app purchase will be rewarded 1 XP and increase in the app sales percentage yields XP bonuses.

%20	%40	%60	%80	%80 +
10 XP	20 XP	30 XP	40 XP	50 XP

Table 5: Earning Selling App Bonuses

Figure 28 and figure 29 shows some basic platform meta-data action.



Name
entity_list

English
Entity List

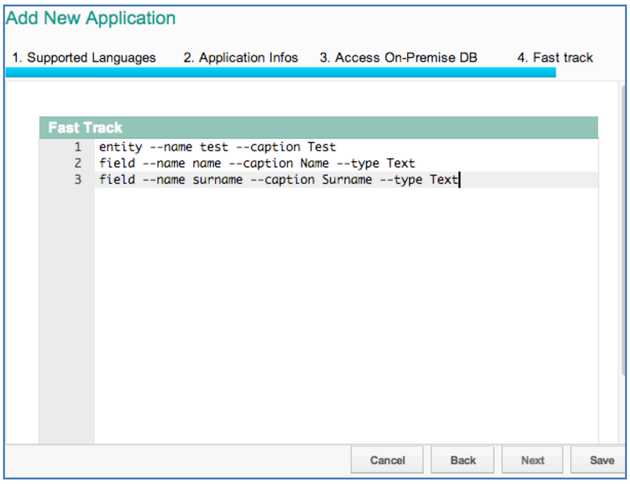
Türkçe
Entity List

Type
Entity List

Cardinality
Many to many

Type of List
t_crm$Contact

Rendering Type
Table Visible On Screen



```
entity --name test --caption Test
field --name name --caption Name --type Text
field --name surname --caption Surname --type Text
```

Figure 28: Create Term that Type is Entity List (Many-to-Many Relation)

Figure 29: Use Fast-Track Editor

Activity	**Required Total Actions for Bronze Badges**	**Required Total Actions for Silver Badges**	**Required Total Actions for Gold Badges**
Total Meta-data Actions	500	1500	2500
Create New App	1	5	10
Create Entity	5	50	100
Create Term	25	250	500
Create Constant	25	250	500
Create Constant Group	25	250	500
Create Transient Field	25	250	500
Use Fast-Track	1	10	20
Create and Use Term that Type is Entity	1		
Create and Use Term that Type is Entity List	1		
Action Rewards	10 x (Bronze Badge + 10 XP)	8 x (Silver Badge + 20 XP)	8 x (Gold Badge + 30 XP)
TOTAL EARNABLE POINTS	**500 XP**		

Table 6: Basic Platform Meta-data Actions

Create / Update / Delete **Metadata- Actions**	**Required Total 10 Actions for Bronze Badges**	**Required Total 100 Actions for Silver Badges**	**Required Total 1000 Actions for Gold Badges**
Entity Actions	1 XP / per Action	1 XP / per 10 Action	1 XP / per 100 Action
Term Actions	1 XP / per Action	1 XP / per 10 Action	1 XP / per 100 Action
Constant Actions	1 XP / per Action	1 XP / per 10 Action	1 XP / per 100 Action
Constant Group Actions	1 XP / per Action	1 XP / per 10 Action	1 XP / per 100 Action
Action Points	4 x 10 XP	4 x 10 XP	4 x 10 XP
Create / Update / Delete **Metadata- Actions**	**Required First Action for Bronze Badges**	**Required Total 10 Actions for Silver Badges**	**Required Total 50 Actions for Gold Badges**
Import Entity	5 XP	1 XP / per Action	1 XP / per 10 Action
Import Constant	5 XP	1 XP / per Action	1 XP / per 10 Action
Use Inheritance	5 XP	1 XP / per Action	1 XP / per 10 Action
WS Message	5 XP	1 XP / per Action	1 XP / per 10 Action
Rest Service	5 XP	1 XP / per Action	1 XP / per 10 Action
Services	5 XP	1 XP / per Action	1 XP / per 10 Action
Report Templates	5 XP	1 XP / per Action	1 XP / per 10 Action
Scheduled Service	5 XP	1 XP / per Action	1 XP / per 10 Action
Action Points	8 x 5 XP	8 x 10 XP	8 x 5 XP
Action Rewards	12 x (Bronze Badge + 10 XP)	12 x (Silver Badge + 20 XP)	12 x (Gold Badge + 30 XP)
TOTAL POINTS	**1000 XP**		

Table 7: CRUD Platform Meta-data Actions

4.1 Pilot Group 2

We tested the proposed scoring model in Hackathon Contest that was held in ITU Çekirdek, 11 teams consisted of 2-3 people chose ImonaCloud platform to develop their projects. All of competitors have knowledge about programming and they have no experience on ImonaCloud platform before. Firstly, we gave a crash course on how to develop apps on ImonaCloud, and then wanted them to take the online course on udemy.com/imonacloud.

This contest was a challenge for us as well; because each team developed different projects and our scoring model tested in such a great organization first time whether the model was viable for that kind of aPaaS platforms.

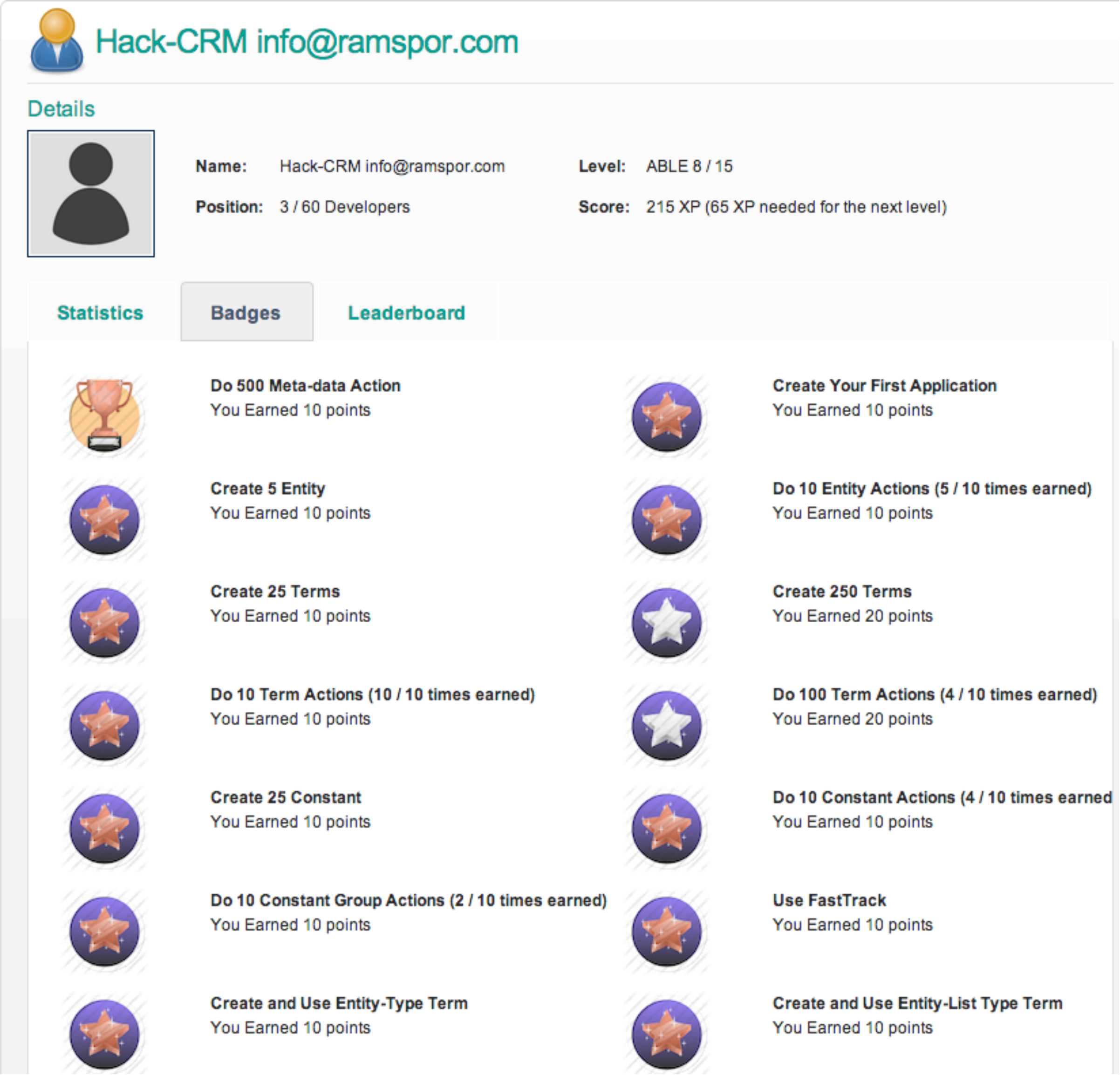
We have observed that,

- During the contest, 2048 meta-data actions were made, 151 screens, 138 entities, 767 fields, 13 web services and 3 scheduled services were created. One team used fast-track editor and 2 team made dependent apps.
- 4 teams tried to develop their projects on the platform but a short time later they gave up due to platform not meet the requirements for their needs.
- 7 teams developed their applications and they finished up their projects in 36 hours.
- According to Hackathon Contest results [16], all 7 team who continued developing their application on ImonaCloud platform, won the competition.

<table>
<tr><td colspan="6">TEAMS THAT DEVELOPED APPLICATIONS
ON
IMONACLOUD PLATFORM</td></tr>
<tr><td colspan="2"><b>RAMSPOR</b></td><td colspan="4"><b>ROCKET</b></td></tr>
<tr><td colspan="2">USER1
215 XP</td><td>USER1
161 XP</td><td colspan="2">USER2
87 XP</td><td>USER3
18 XP</td></tr>
<tr><td colspan="2"><b>AGHLAB</b></td><td colspan="4"><b>FCLABS</b></td></tr>
<tr><td>USER1
123 XP</td><td>USER2
55 XP</td><td colspan="2">USER1
125 XP</td><td colspan="2">USER2
121 XP</td></tr>
<tr><td colspan="2"><b>DR. CLOUD</b></td><td colspan="2"><b>FRAMETECH</b></td><td colspan="2"><b>DU COMPUTER</b></td></tr>
<tr><td colspan="2">USER1
96 XP</td><td colspan="2">USER1
82 XP</td><td colspan="2">USER1
54 XP</td></tr>
<tr><td colspan="6">TEAMS
THAT GAVE UP DEVELOPING APPLICATIONS
ON
IMONACLOUD PLATFORM</td></tr>
<tr><td colspan="2"><b>HALLEDERIZ</b></td><td colspan="4"><b>MONSTERS</b></td></tr>
<tr><td colspan="2">USER1
59 XP</td><td colspan="4">USER1
47 XP</td></tr>
<tr><td colspan="2"><b>KAFEIN</b></td><td colspan="4"><b>SOB</b></td></tr>
<tr><td colspan="2">USER1
40 XP</td><td colspan="4">USER1
11 XP</td></tr>
</table>

Table 8: Hackathon Contest Competitors

As it shown in table, after categorizing groups, it is seen that developers who got lower points were the same developers who gave up developing applications on ImonaCloud platform. Except Ramspor Team, all teams who work with one account earned lower points in overall. On the contrary working with multiple accounts teams gained more points totally. As it is mentioned before, all 7 team in the first group who continued developing their application on ImonaCloud platform also won the competition. It can be said that, the model we proposed viable for that kind of aPaaS platorms.


Hack-CRM info@ramspor.com

Details

Name: Hack-CRM info@ramspor.com

Level: ABLE 8 / 15

Position: 3 / 60 Developers

Score: 215 XP (65 XP needed for the next level)

Do 500 Meta-data Action
You Earned 10 points

Create Your First Application
You Earned 10 points

Create 5 Entity
You Earned 10 points

Do 10 Entity Actions (5 / 10 times earned)
You Earned 10 points

Create 25 Terms
You Earned 10 points

Create 250 Terms
You Earned 20 points

Do 10 Term Actions (10 / 10 times earned)
You Earned 10 points

Do 100 Term Actions (4 / 10 times earned)
You Earned 20 points

Create 25 Constant
You Earned 10 points

Do 10 Constant Actions (4 / 10 times earned
You Earned 10 points

Do 10 Constant Group Actions (2 / 10 times earned)
You Earned 10 points

Use FastTrack
You Earned 10 points

Create and Use Entity-Type Term
You Earned 10 points

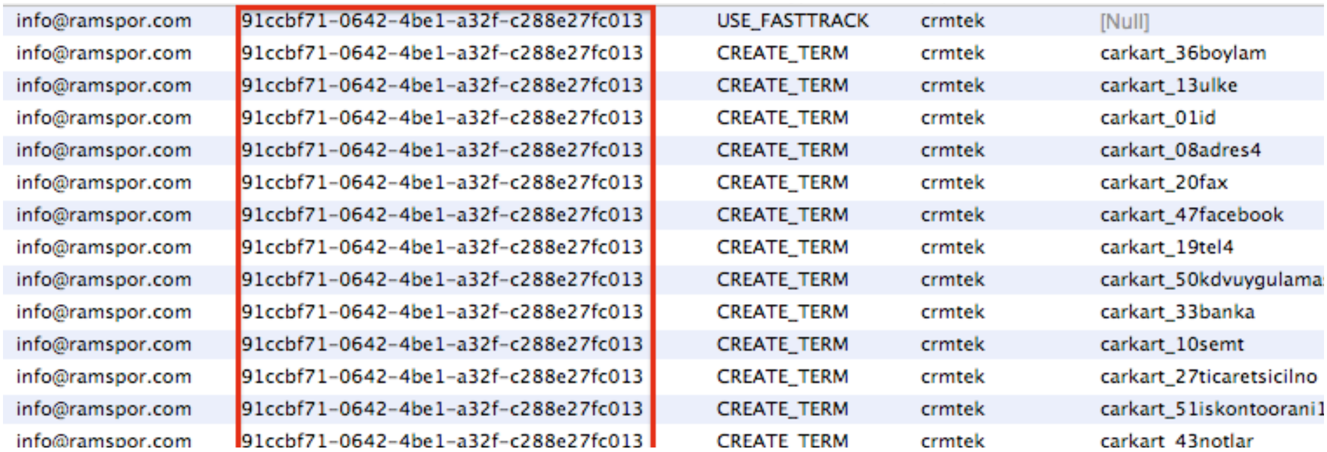
Create and Use Entity-List Type Term
You Earned 10 points

Figure 30: Ramspor Earned Badges

A close analysis of Ramspor team meta-data actions shows that they earned 160 points only from by completing 14 / 62 challenges. In the platform there were 12 bronze badges pre-defined and Ramspor team got all of bronze badges. The big difference actions from other teams are that they created over 500 meta-data action in 2 days and they used fast-track editor.

Fast-Track editor is an ImonaCloud feature tool that you can prepare requirement fields, entities for your application and execute your scripts faster. It saves considerable time on developing applications. Ramspor developed their application by using the fast-track editor tool. This feature may lead to abuse the scoring system. We are aware of it and thinking a solution to make another point system for whom to use fast-track editor. It is the only place that more than 2 actions made at one time. That is, when fast-track is used, we bind multiple actions with "uuid" and add extra action to "Use Fast-Track" in database. And we know that these all actions made with fast-track tools. For now we did not close this because this is platform's feature but we keep tracking meta-actions made with fast-track tool.

One of the precautions that we take is that challenges were separated in three categories: bronze, silver and gold. As it is known, gold challenges require more meta-data actions than the other categories. So, developers will spend more time to earn XP.



info@ramspor.com	91ccbf71-0642-4be1-a32f-c288e27fc013	USE_FASTTRACK	crmtek	[Null]
info@ramspor.com	91ccbf71-0642-4be1-a32f-c288e27fc013	CREATE_TERM	crmtek	carkart_36boylam
info@ramspor.com	91ccbf71-0642-4be1-a32f-c288e27fc013	CREATE_TERM	crmtek	carkart_13ulke
info@ramspor.com	91ccbf71-0642-4be1-a32f-c288e27fc013	CREATE_TERM	crmtek	carkart_01id
info@ramspor.com	91ccbf71-0642-4be1-a32f-c288e27fc013	CREATE_TERM	crmtek	carkart_08adres4
info@ramspor.com	91ccbf71-0642-4be1-a32f-c288e27fc013	CREATE_TERM	crmtek	carkart_20fax
info@ramspor.com	91ccbf71-0642-4be1-a32f-c288e27fc013	CREATE_TERM	crmtek	carkart_47facebook
info@ramspor.com	91ccbf71-0642-4be1-a32f-c288e27fc013	CREATE_TERM	crmtek	carkart_19tel4
info@ramspor.com	91ccbf71-0642-4be1-a32f-c288e27fc013	CREATE_TERM	crmtek	carkart_50kdvuygulama
info@ramspor.com	91ccbf71-0642-4be1-a32f-c288e27fc013	CREATE_TERM	crmtek	carkart_33banka
info@ramspor.com	91ccbf71-0642-4be1-a32f-c288e27fc013	CREATE_TERM	crmtek	carkart_10semt
info@ramspor.com	91ccbf71-0642-4be1-a32f-c288e27fc013	CREATE_TERM	crmtek	carkart_27ticaretsicilno
info@ramspor.com	91ccbf71-0642-4be1-a32f-c288e27fc013	CREATE_TERM	crmtek	carkart_51iskontoorani1
info@ramspor.com	91ccbf71-0642-4be1-a32f-c288e27fc013	CREATE_TERM	crmtek	carkart_43notlar

Figure 31: Fast-Track multi actions at one time

Chapter 5

Conclusion

In this thesis, we integrated game mechanics like points, level, badges and leaderboards on ImonaCloud platform aiming to encourage developers to improve their engagement and effectiveness on their platform practices. We designed the point system and increased the points gained according to the difficulty level of the challenge.

We tested our scoring model in Hackathon Contest that was held in ITU Çekirdek, 11 teams consisted of 2-3 people developed their projects by using ImonaCloud platform. By looking their meta-data actions and scores they got, we can say that, our scoring model is viable for that kind of aPaaS platforms. Our gamification model may help not only the developers but for the novice programmers to learn the platform faster and become more engaged. This is subject to further investigation as a future research topic.

Another future work, with the help of SonarQube's open source inspection code [18], developers' written scripts will also evaluated and scored. And this score will be added to developers' earned points.

Also, we will extend and improve our model on different experience groups, including both non-coders and coders, and we will track their progress on the platform a long period of time, monitoring their actions on the platform, and scoring them against these actions.

One of the other future works is to expose gamification APIs to the ImonaCloud developer community so that they can leverage the API to combine their apps with gamification principles and create their own scoring and engagement model.

References

[1] Zichermann, Gabe; Cunningham, Christopher (August 2011). "Introduction". Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps (1st ed.). Sebastopol, California: O'Reilly Media. p. xiv. ISBN 1449315399. Retrieved 2012-12-10.

[2] Deterding, Sebastian; Dixon, Dan; Khaled, Rilla; Nacke, Lennart. (2011), "From Game Design Elements to Gamefulness: Defining "Gamification" " Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments. ACM, 2011"

[3] Gartner, (2012), "Gartner Reveals Top Predictions for IT Organizations and Users for 2013 and Beyond", http://www.gartner.com/newsroom/id/2211115 [Online; accessed 22-August-2014].

[4] Hamari, J., Koivisto, J., & Sarsa, H. (2014), "Does Gamification Work? – A Literature Review of Empirical Studies on gamification. In proceedings of the 47th Hawaii International Conference on System Sciences, Hawaii, USA. "

[5] Deloitte, (2013), "Gamification Goes To Work", http://www.deloitte.com/assets/Dcom-UnitedStates/Local%20Assets/Documents/Consulting/Technology/TechTrends13/us_cons_TT2013Gamification_Infographic_111913.pdf, [Online; accessed 19-August-2014].

[6] Fauscette, Michael, (2013), "Is There Value In Enterprise Gamification? Part Three", http://www.mfauscette.com/software_technology_partn/2013/04/is-there-value-in-enterprise-gamification-part-three.html, [Online; accessed 10-August-2014].

[7] Personal Metrics, (2011), "Using Personel Metrics to drive workplace performance", http://www.personalmetrics.us/2011/07/using-personal-metrics-to-drive-workplace-performance/, [Online; accessed 20-August-2014].

[8] Strohmeyer, Robert (Aug, 2013), "Gamification: Using play to motivate employees and engage customers", http://www.pcworld.com/article/2047564/gamification-using-play-to-motivate-employees-and-engage-customers.html

[9] Smolander K, (2010) , “Platform as a Service – new opportunities for software development companies”, Lappeenranta University of Technology

[10] Velte T., Velte A., Elsenpeter R, (2009), “Cloud Computing, A Practical Approach”, McGraw-Hill Osborne Media, US,

[11] Mather T., Kumaraswamy S., Latif S., “Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance (Theory in Practice)”, O’Reilly Media, US. (2009).

[12] “Public vs Private Cloud Computing”, http://www.onlinetech.com/resources/wiki/cloud-computing/public-vs-private-cloud-computing?__hstc=43953530.723a0d2da71e6f91fa3eb5826752a830.1408848626122.1408848626122.1408848626122.1&__hssc=43953530.1.1408848626123&__hsfp=676482550, [Online; accessed 23-August-2014].

[13] Gens, Frank; December 2011, “TOP 10 PREDICTIONS”. IDC #231720, Volume: 1

[14] Massimo Pezzini and Benoit J, (2011) “ Integration Platform as a Service: Moving Integration to the Cloud”. Gartner RAS Core Research Note G00210747, 7 March 2011, pp. 2-9.

[15] Sabandam, Suresh; (2012), “Navigating the PaaS Maze”. [Online; accessed 23-August-2014].

http://sandhill.com/article/navigating-the-paas-maze/, [Online; accessed 23-August-2014].

[16] OrangeScape, (2011), “Demystifying PaaS Landscape – Fishing in muddied waters of Cloud”, http://blog.orangescape.com/2011/10/18/1265/, [Online; accessed 19-August-2014].

[17] Sirinavasan P., (2012), Cloud Epxo Article: “Inspecting the PaaS Landscape where Azure fits”, http://cloudcomputing.sys-con.com/node/2211937, [Online; accessed 22-August-2014].

[18] Itü Çekirdek, (2014), “Bulut Bizden Hackathon Sonuçları Açıklandı”, http://www.itucekirdek.com/haberler/bulut-bizden-hackathon-sonuclari-aciklandi, [Online; accessed 21-August-2014].

[19] SonarQube - http://www.sonarqube.org/, [Online; accessed 23-August-2014].

Curriculum Vitae

Mustafa Volkan Gürol was born in November 13th, 1987, in Istanbul. He received his BS in Computer Engineering and Statistics and Computer Science in 2011 at Kadir Has University. Since 2012 he has been working as a software engineer at Imona Technologies Ltd.