

클라우드컴퓨팅 서비스 사례연구

Case Study of Mobile Cloud Computing Service

시드니 망고예¹⁾, 심동희²⁾

Sidney Mangoye , Donghee Shim

요 약

클라우드컴퓨팅은 새로운 정보기술환경으로 발전하고 있으며 우수한 IT기업들의 클라우드컴퓨팅서비스가 제공되고 있다. 개인창업자나 중소기업에서는 PaaS(Platform as a Service)를 이용하여 빠른 시간에 쉽게 기업의 정보시스템을 구축할 수 있게 되었다. 또한 스마트폰의 폭발적인 사용과 더불어 모바일클라우드컴퓨팅도 소비자에게는 편리한 IT환경을 제공하고 있다. 본 연구에서는 이러한 모바일클라우드컴퓨팅 서비스와 BaaS(Backend as a Service)를 이용하여 간단한 시스템을 구현한 사례연구다. 그리하여 스마트폰에서 로그인 을 하여 텍스트나 이미지 자료를 업로드할 수 있게 한다. 이러한 모바일클라우드컴퓨팅서비스를 이용하면 개인창업자들이 정보시스템개발에 소요되는 시간을 최소화할 수 있음을 보이고 있다.

핵심어: 클라우드컴퓨팅, 모바일 클라우드컴퓨팅, 백엔드서비스, 모바일 응용

Abstract

Cloud computing has been developed as a new IT environments and cloud computing services have been provided by many IT companies. Startup companies and small businesses can establish an information system using PaaS easily and quickly. Moreover mobile cloud computing provide a convenient IT environments to customers with an explosive use of smartphone. This research is a case study which implement a simple system using both mobile cloud computing service and BaaS(Backend as a Service). After user logs in using smartphone, text or image data can be uploaded. Startup companies can save the time in developing an information system using these mobile cloud computing services.

Keyword: Cloud Computing, Mobile Cloud Computing, Backend as a Service, Mobile Application

1) Department of Computer Science & Engineering, Graduate School , Jeonju University (First author)

2) Department of Computer Science & Engineering, Jeonju University (Corresponding author)

Received(May 10. 2018), Review (May 29. 2018), Accepted(June 30. 2018)

1. 서론

클라우드컴퓨팅(CC:Cloud Computing)은 최소의 관리노력으로, 빠르게 제공될 수 있는, 설정가 능한 컴퓨팅자원에게 네트워크를 통하여 언제 어디서나 편리하게 필요시에 접근을 가능하게 해 주는 모델이다[1]. CC는 가상화, 분산시스템 그리고 네트워킹과 같은 서로 다른 컴퓨팅패러다임 의 진화적인 융합인 것이다. 이런 CC도 단점이 있는 바 보안문제 등에 대한 포괄적인 분석도 제 시된 바 있다[2].

최근에는 다양한 종류의 어플리케이션들이 제공되면서 스마트폰이 주요 컴퓨팅장치로 사용되 고 있다. 이러한 이용은 현대사회에 점점 더 큰 영향을 주게되어, 기업이나 조직에서는 IT 인프 라구조에 모바일부분도 모두 통합해주는 동향이 있다. 그러나 모바일장치는 데스크탑 호스트와 비교하여 성능, 환경, 보안 등의 면에서 취약하다[3]. 그래서 모바일클라우드컴퓨팅(MCC:Mobile Cloud Computing)은 이러한 단점을 어느 정도 극복할 수 있도록 CC를 모바일환경으로 통합을 해가고 있다[4][5]. MCC의 주요 이슈는 프라이버시와 자료보안, 구조와 모델 그리고 인프라구조 와 네트워크로 나뉘어져 연구되고 있다[5].

이 논문에서는 MCC의 스택과 BaaS(Backend as a Service)[6]를 이용한 클라이언트-서버 통신 을 이용한 사례연구를 한다. 사용자가 스마트폰에서 로그인을 한 후, 텍스트자료나 이미지자료를 업로드하면 이를 BaaS를 이용하여 저장하고 스마트폰에 이 내용을 다시 표시해주는 간단한 시 스템이다.

본 논문의 구성은 다음과 같다. 2장에서는 클라우드 컴퓨팅 관련 이론을 살펴보고, 3장에서는 시스템의 설계와 구현을 설명하며 4장에서는 결론을 기술한다.

2. 클라우드컴퓨팅 관련 이론

2.1 클라우드컴퓨팅

CC는 컴퓨팅서비스 배포, 편리한 사용 등을 효율적으로 저렴한 비용으로 가능하게 해준다[1]. 미국연방표준기술원(NIST:American National Institute of Standards and Technology)에 따르면

CC는 수요에 따른 서비스, 자원의 풀링, 광역네트워크접근, 높은 탄력성, 측정가능한 서비스의 특성을 가지고 있다[6]. CC는 사용특성에 따라 사설클라우드(Public Cloud), 공공클라우드(Public Cloud), 커뮤니티클라우드(Community Cloud) 그리고 혼합클라우드(Hybrid Cloud) 등으로 분류될 수 있다[6].

2.2 클라우드컴퓨팅 서비스

IaaS 모델은 서비스형태에 따라 SaaS(Software as a Service), PaaS(Platform as a Service) 그리고 IaaS(Infrastructure as a Service)로 분류된다[6]. 또한 다른 관점에서 XaaS(Anything as a Service, or Anything as a Service), IDaaS(Identity as a Service), BaaS(Back-end as a Service)도 있다. 그리고 XaaS는 DRaaS(Disaster Recovery as a Service), MaaS(Management as a Service) 그리고 SECaaS(Security as a Service)을 포함한다[6]. 새로운 기술과 비즈니스모델의 등장에 따라 XaaS는 여전히 여러 유형으로 발전하고 있다. 이러한 분류중 BaaS는 PaaS의 특별한 형태로서 어플리케이션 개발자들에게 설정이 가능하고 커스터마이징이 가능한 백엔드(Backend) 솔루션을 완전하게 제공한다[7]. 즉 BaaS는 서버 측에서의 데이터베이스 저장 및 관리, 저장, 어플리케이션과의 메시지 송수신 등과 같은 설정가능한 서비스를 어플리케이션과 연결해주는 방법을 제공한다. 그리하여 어플리케이션 개발자들이 더 빠르고 효율적으로 모바일 어플리케이션들 구축하고 배포할 수 있도록 한다. Microsoft, Amazon, Google, IBM 그리고 Apple 등을 비롯한 우수한 클라우드 컴퓨팅 서비스업체들이 있다. 이들은 차별성이 있는 서비스를 제공하기 위하여 노력중이다.

2.3 모바일 클라우드컴퓨팅

MCC의 전형적인 이단(2-tiers) 구조가 소개되었다[8]. 이 이단구조에서 모바일 장치들은 베이스 스테이션을 통해 모바일 네트워크에 연결된다. 그러나 일부에서는 Cloudlet 등을 이용해 삼단(third-tier)를 취하여 네트워크의 신뢰성 문제를 해결해야 한다고 강조한다[9]. 삼단구조에서는 모바일장치와 클라우드 네트워크 사이에 CDN(Contents Delivery Network) 이나 Repetitor와 같은 기능을 수행하는 층을 추가한다[9]. Cloudlet은 서비스전달시에 높은 대역폭을 제공하고 또한 지연 접근을 줄여주는 마이크로 클라우드다[10].

최근 MCC를 사용하는 많은 모바일 어플리케이션들이 개발되고 있다[11][12]. 이들은 CC 접근 법만이 제공할 수 있는 기능들을 MCC에서 성공적으로 제공함으로써 관련 시장에서 중요한 영향력을 보이고 있다. 이들의 주요분야는 음성인식과 텍스트의 음성변환, 모바일 학습, 위치 및 내용물 탐지서비스, 모바일 상거래, 모바일 헬스케어, 모바일 게임 등이다.

모바일 컴퓨팅은 이동성에 내재된 몇가지 제약성을 가지고 있다. 이 모바일컴퓨팅에 CC를 적용하여 하드웨어의 제한, 배터리 문제 그리고 계산능력 등과 같은 주요 제약점 대부분을 극복할 수 있다. 뿐만 아니라 확장성, 향상된 저장공간, 데이터와 어플리케이션의 신뢰성, 하나의 시스템을 분리해 다수고객 수용능력, 유연한 통합 등의 장점을 갖게된다.

표 1에는 모바일 플랫폼을 iOS와 Android로 구분하여 MCC에서 사용되는 기술요소들을 대비하여 나타냈다. 표 1에서 HTTP 서비스 라이브러리는 CC 서비스에 쉽게 접근할 수 있도록 라이브러리를 제공하는 써드파티그룹에서 운영하는 것이다. 여기에 나타낸 BaaS의 역할은 앞 소절에서 설명한 바와 같다.

[표 1] 모바일 플랫폼 비교분석
 [Table 1] Comparison of Mobile Platforms

구분	iOS	Android
개발 언어	Swift, Objective-C	Java
IDE	Xcode	Android Studio, Eclipse ADT
대상 모바일	아이폰, 아이패드, 애플TV, 애플시계, 맥OSX	안드로이드폰/태블릿/Wear
HTTP 서비스 라이브러리	Alamofire 등	Volley 등
버전제어	Git, GitHub 등	
BaaS	Firebase, Parse, Google Comute Engine, Apache Usergrid 등	
	CloudKit	
환경	Mac OS X	MS-Windows, Linux, Mac OS C

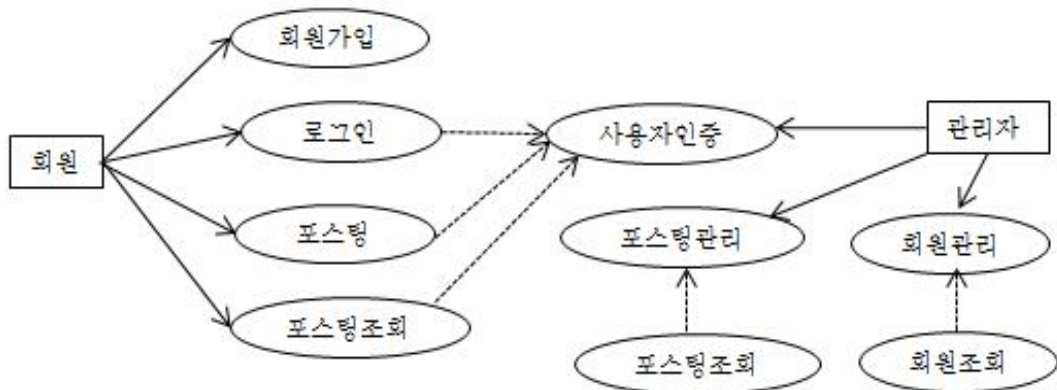
3. 시스템 설계 및 구현

3.1 시스템 개요

우리가 개발하는 JJConnect는 BaaS를 이용하여 아주 쉽게 시스템을 구축할 수 있는 것을 보여주기 위한 프로토타입의 간단한 시스템이다. 먼저 스마트폰을 통하여 사용자의 로그인을 받는데 사용자가 직접 회원가입을 하거나 또는 페이스북 계정을 이용하여 로그인 할 수 있다. 그리고 사용자가 올리는 텍스트나 이미지를 제목과 함께 클라우드서버에 저장하고 이를 스마트폰에 보여주는 간단한 시스템이다. 생성되는 회원정보와 업로드되는 텍스트와 이미지자료는 모두 BaaS의 서비스를 받아서 처리되고 사용자는 제목을 통하여 조회할 수 있다.

3.2 시스템 UML 모델링

JJConnect에 대하여 UML 모델링을 통한 시스템설계를 한다. 먼저 Fig. 1에 유스케이스 다이어그램을 나타냈다.



[그림 1] 유스케이스 다이어그램 (점선은 포함관계)

[Fig. 1] Usecase Diagram

Table 2에는 유스케이스식별자목록을 나타낸다. 식별자에서 A는 관리자, M은 회원을 나타낸다.

[표 2] 유스 케이스 식별자 목록
[Table 2] Usecase Identifier List

식별자	행위자	설명
UC-M1	회원	회원가입
UC-M2	회원	로그인
UC-M3	회원	포스팅
UC-M4	회원	포스팅조회
UC-A1	관리자,회원	사용자인증
UC-A2	관리자	회원관리
UC-A3	관리자	포스팅관리

Table 3부터 9까지는 Table 2에 나타난 유스케이스식별자목록 각각에 대한 유스케이스시나리오다.

[표 3] UC-M1의 유스 케이스 시나리오
[Table 3] Usecase Scenario for UC-M1

<p>1.개요:회원이 스마트폰을 통하여 회원가입을 함</p> <p>2.관계: 시작자:회원</p> <p>3.기본흐름</p> <ul style="list-style-type: none"> - 사용자가 사용자명과 암호를 입력함 <p>(사용자명은 기존 등록된 다른 사용자명과 중복이 없어야함)</p>
--

[표 4] UC-M2의 유스 케이스 시나리오
[Table 4] Usecase Scenario for UC-M2

<p>1.개요:회원이 스마트폰을 통하여 로그인</p> <p>2.관계: 시작자:회원</p> <p>3.기본흐름</p> <ul style="list-style-type: none"> - 회원이 사용자명과 암호로 로그인

[표 5] UC-M3의 유스 케이스 시나리오
[Table 5] Usecase Scenario for UC-M3

1.개요:회원이 스마트폰을 통하여 포스팅
2.관계:
- 시작자:회원
- 선행조건: 회원이 로그인한 후
3.기본흐름
- 텍스트나 이미지를 포스팅하는데, 제목을 입력하고 텍스트의 경우 80바이트이하, 이미지의 경우 10M 이하의 파일

[표 6] UC-M4의 유스 케이스 시나리오
[Table 6] Usecase Scenario for UC-M4

1.개요:회원이 스마트폰을 통하여 포스팅조회
2.관계:
- 시작자:회원
- 선행조건: 회원이 로그인한 후
3.기본흐름
- 텍스트나 이미지의 제목을 입력하여 조회하거나 제목을 모르면 포스팅일시로 확인하여 조회할 자료를 선택

[표 7] UC-A1의 유스 케이스 시나리오
[Table 7] Usecase Scenario for UC-A1

1.개요:사용자 인증
2.관계:
- 시작자:회원, 관리자
3.기본흐름
- 회원이 입력한 사용자명과 암호로 일치여부를 확인함

[표 8] UC-A2의 유스 케이스 시나리오
[Table 8] Usecase Scenario for UC-A2

1.개요:회원관리
2.관계:
- 시작자:관리자
3.기본흐름
- 관리자가 회원의 사용자명과 암호를 체크함

[표 9] UC-A3의 유스 케이스 시나리오

[Table 9] Usecase Scenario for UC-A3

1.개요:포스팅관리

2.관계:

- 시작자:관리자

3.기본흐름

- 관리자가 회원의 입력한 텍스트자료와 이미지자료를 관리함

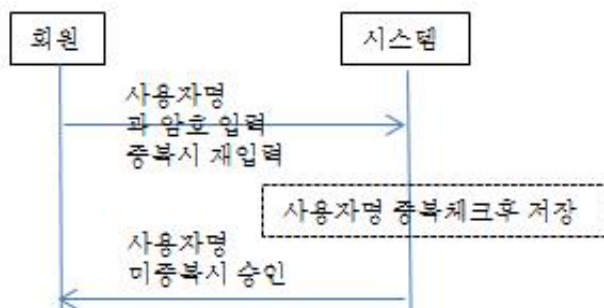
Fig. 2에는 클래스다이어그램을 나타낸다.



[그림 2] 클래스 다이어그램

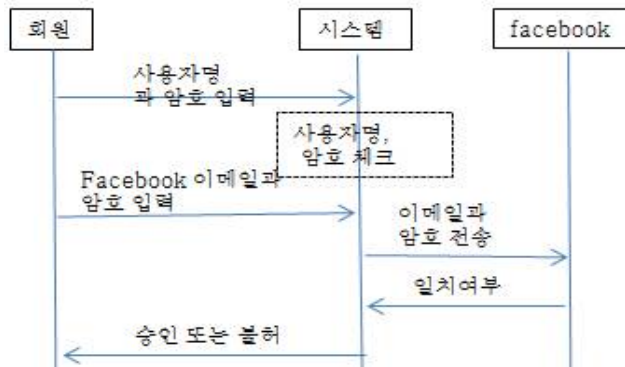
[Fig. 2] Class Diagram

Fig. 3부터 Fig. 9에는 유스케이스 시나리오 각각에 대한 시퀀스다이어그램을 나타내고 있다.

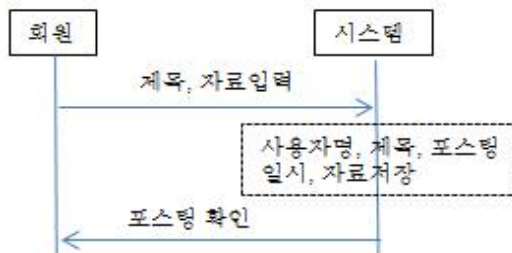


[그림 3] UC-M1의 시퀀스 다이어그램

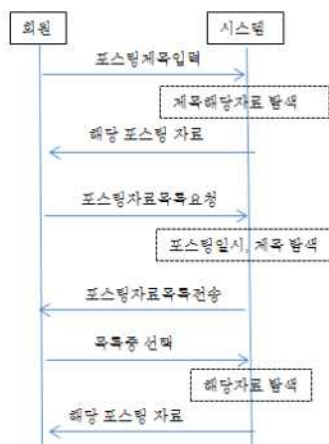
[Fig. 3] Sequence Diagram for UC-M1



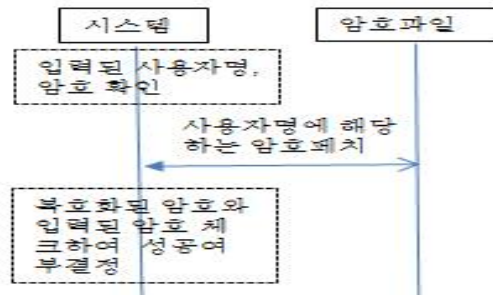
[그림 4] UC-M2의 시퀀스 다이어그램
 [Fig. 4] Sequence Diagram for UC-M2



[그림 5] UC-M3의 시퀀스 다이어그램
 [Fig. 5] Sequence Diagram for UC-M3



[그림 6] UC-M4의 시퀀스 다이어그램
 [Fig. 6] Sequence Diagram for UC-M4



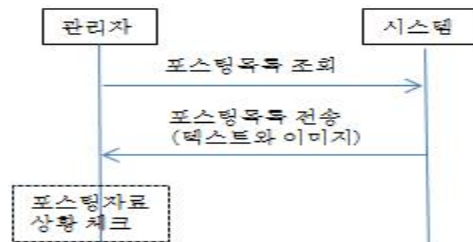
[그림 7] UC-A1의 시퀀스 다이어그램

[Fig. 7] Sequence Diagram for UC-A1



[그림 8] UC-A2의 시퀀스 다이어그램

[Fig. 8] Sequence Diagram for UC-A2



[그림 9] UC-A3의 시퀀스 다이어그램

[Fig. 9] Sequence Diagram for UC-A3

3.3 시스템 구현 환경

먼저 모바일 어플리케이션은 애플의 iOS로 국한하여 구현하며, IDE로 Xcode를 이용하고 Swift 언어를 이용한다[13][14][15]. 이 어플은 JConnect의 프론트엔드(Frontend) 서비스를 담당한다. BaaS로는 Firebase[16]를 이용한다. Firebase 프레임워크는 백엔드(Backend) 서비스에 저장된 데이터의 처리를 담당한다. 그리고 HTTP 서비스 라이브러리로 Swift로 작성된 Alamofire[17]를 사

용한다. Alamofire는 URL, JSON 파라미터 인코딩, 파일과 데이터의 업로드, 다중형태데이터, HTTP 응답체크, NSURLConnection을 이용한 인증 등과 같은 특성을 제공한다. 또한 이미지를 저장하기 위하여 Firebase를 통하여 ImageHack[18]을 사용한다. 그래서 텍스트데이터는 Firebase를 가져오며, 이미지는 다른 클라우드서비스제공업체의 ImageHack에서 Firebase를 통하여 가져온다.

그리고 애플의 iCloud 서버에 대한 통합적인 접근을 위하여 CloudKit[19]를 사용하는데 이는 앞서 Table 1에 나타낸 바와 같이 iOS장치에만 지원된다.

3.4 시스템 구현

여기서 구현한 JConnect는 Alamofire 프레임워크를 이용하여 모든 네트워크통신을 처리한 소켓기반 네트워크 응용시스템이다. 새로운 포스트 업로드에서 Alamofire는 JSON 데이터에 이미지를 압축하고 다중형태데이터로 ImageHack의 클라우드서버에게 보낸다. 또한 ImageHack에 의하여 생성된 이미지 URL을 가져오고 이 이미지 URL은 Firebase 데이터베이스에 저장된다. 구현에서 버전제어를 위하여 GitHub[20]을 사용했다.

클라이언트 어플은 Model-View-Controller[21]에 기반한 iOS 프레임워크를 이용하였다. 그리하여 모델로는 데이터서비스, 상수 그리고 포스트 3개를 생성했다. 그리고 로컬데이터를 처리하는 데이터서비스 모델에 대응되는 DataService.swift, 사용자명이나 오류코드 같은 상수를 처리하는 Constants.swift, 포스트를 처리하는 Post.swift를 구현했다. 그리고 3개의 컨트롤러를 생성했는데 전체를 관장하는 뷰컨트롤러, 포스트를 관장하는 포스트셀 컨트롤러, 백엔드와 컨트롤하는 FeedVC 컨트롤러이다. 이 3 컨트롤러를 처리하는 ViewController.swift, PostCell.swift 그리고 FeedVC.swift를 구현했다. 그리고 전체를 위한 1개의 스토리보드를 생성했다. 뷰에는 메인뷰, 로그인/서명뷰 그리고 론치스크린뷰를 생성하였다.

4. 결론

본 논문에서 개발한 JConnect는 네트워크를 통해 데이터를 보내고 받을 수 있는 모바일 응용이다. 그리고 클라우드에서 실시간으로 자료를 저장하고 동기화하기 위하여 클라우드서비스를 설정하기 위하여 BaaS로 Firebase를 이용하였다. 어플은 설계했던 모든 기능을 BaaS를 통하여

성공적으로 수행하였다. 클라우드 백엔드를 이용하여 사용자인증과 데이터송수신을 안전한 방법으로 수행하며 클라우드 자원에 접근한다. 써드파티의 API를 통하여 사용자인증을 외부에 저장된 페이스북의 SNS를 통한 로그인도 허용하고 있다. 이러한 접근방법은 모바일클라우드 서비스를 개발하는 다른 프로젝트에서도 활용될 수 있다.

이 논문의 결과는 BaaS의 편의성을 평가할 수 있도록 한다. BaaS를 통하여 개발에 소요되는 시간을 절약하고, 복잡성을 줄여주고, 이동성을 증대시켜준다. 또한 상대적으로 추가적인 많은 작업 없이 이는 다른 플랫폼에서도 사용될 수 있다. 스타트업기업이나 중소기업에서 정보시스템을 개발하는 경우 BaaS는 MCC를 쉽게 개발할 수 있도록 한다. 대부분의 경우 BaaS에서는 높은 수준의 어플이 아닌 경우 비용 발생이 전혀 없어서 초기개발비용을 크게 줄일 수 있을 뿐만 아니라 유지보수비용도 획기적으로 줄일 수 있다.

References

- [1] Huan Ma Gaofeng Shen, Ming Ched and Jianwei Zhang, “Technologies based on Cloud Computing Technology”, *Advanced Science and Technology Letters Vol.82 ,SoftTech* (2015):1-5
- [2] M. Al Morsy, J. Grundy and I. Müller, “An analysis of The Cloud Computing Security Problem”, In *Proceedings of APSEC 2010, Cloud Workshop, Sydney, Australia*, (2010).
- [3] M. Satyanarayanan, “Fundamental Challenges in Mobile Computing”, in *Proceedings of the 5th annual ACM symposium on Principles of distributed computing*, (1996):1-7
- [4] A. M. S. Zunaitha Sultana, L. Clara Mary, A. Sangeetha, “Swot Analysis of Mobile Cloud Computing”, *International Journal of Scientific and Research Publications*, Vol. 3, Issue 8, (2013)
- [5] M. Satyanarayanan, “Mobile Computing: the Next Decade”, in *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*, (2010)
- [6] NIST Cloud Computing Reference Architecture, National Institute of Standards and Technology, <https://www.nist.gov/document-4641/> Retrieved Dec. 27. (2017)
- [7] J. Y. Lee, B. S. Cho, “An Analysis on Cloud Mobile BaaS Market Perspectives”, *Electronics and telecommunications trends*, Vol.30 No. 4, (2015):191-202
- [8] M. R. Prasad, J. Gyani and P. R. K. Murti, “Mobile Cloud Computing: Implications and Challenges“, *Journal of Information Engineering and Applications*, vol. 2, no. 7, (2012)
- [9] M. R. Prasad, J. Gyani and P. R. K. Murti, “Mobile Cloud Computing: Implications and Challenges”, *Journal of Information Engineering and Applications*, vol. 2, no. 7, (2012)
- [10] MSR’s Paramvir Bahl et al. “The Case for VM- based Cloudlets in Mobile Computing”, *IEEE’s Pervasive Computing journal*, (2009)
- [11] Pragaladan. R1, Leelavathi M., “A Study of Mobile Cloud Computing And Challenges”, *International Journal of Advanced Research in Computer and Communication Engineering* Vol. 3, Issue 7, (2014)
- [12] Elizabeth Halash, “Mobile Cloud Computing: case studies”, *Wayne State University*, (2010)
- [13] Xcode <https://developer.apple.com/xcode/> Retrieved Jan. 03 (2018)
- [14] *The Swift Programming Language (Swift 2.1)*, Apple Inc., (2015)
- [15] *iOS 9 and Swift 2: From Beginner to Paid Professional*, Mark Price, <http://www.udemy.com> Retrieved Jan. 04. (2018)

- [16] Firebase documentation, <https://www.firebase.com/docs/> Retrieved Jan. 05. (2018)
- [17] Alamofire Framework documentation, <https://github.com/Alamofire/Alamofire> Retrieved Jan. 08. (2018)
- [18] ImageHack API documentation, <https://code.google.com/p/imageshackapi/> Retrieved Jan. 09. (2018)
- [19] CloudKit <https://en.wikipedia.org/wiki/CloudKit> Retrieved Jan. 11. (2018)
- [20] GitHub, <https://github.com> Retrieved Jan. 15. (2018)
- [21] Apple Developer forum, <https://forums.developer.apple.com/> Retrieved Jan. 18. (2018)