□ Portfolio

Park Minchul

Contents

Intro

- Park Minchul
  - 1989.02.20
- Technical skill
  - Functional Programming
  - Concurrent Programming
  - Cloud
  - Micro Service
- career
  - Kakao
    - 2014. 09 ~ now
  - Hanwha Techwin
    - 2014. 01 ~ 2016. 9
- Homepage: https://knightpop.github.io/home/
- Blog : http://project-ktz.tistory.com/
- Github : https://github.com/knightpop

kakao

Web frontend using react

- Programming Environments
  - Language : javascript es6
  - Framework : React, Redux

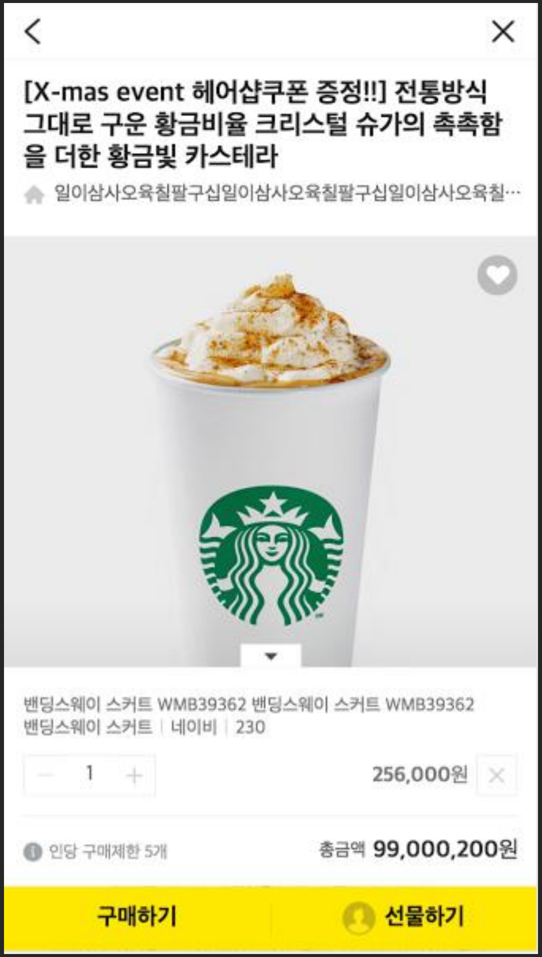
- Project Goal
  - Develop Web App frontend for kakao commerce(gift)
  - Renewal page to replace legacy code(Angular JS 1) with React

- Project Result
  - Best tab page
    - Develop new Best Tab web page to launch new service
  - Renewal Order Project
    - Develop new Item Detail page to enhance UX

- Project Feature
  - Use React, Redux, es6 to Develop modularized front page

☐ Page Views

Item Detail Page



[X-mas event 헤어샵쿠폰 증정!!] 전통방식 그대로 구운 황금비율 크리스털 슈가의 촉촉함을 더한 황금빛 카스테라

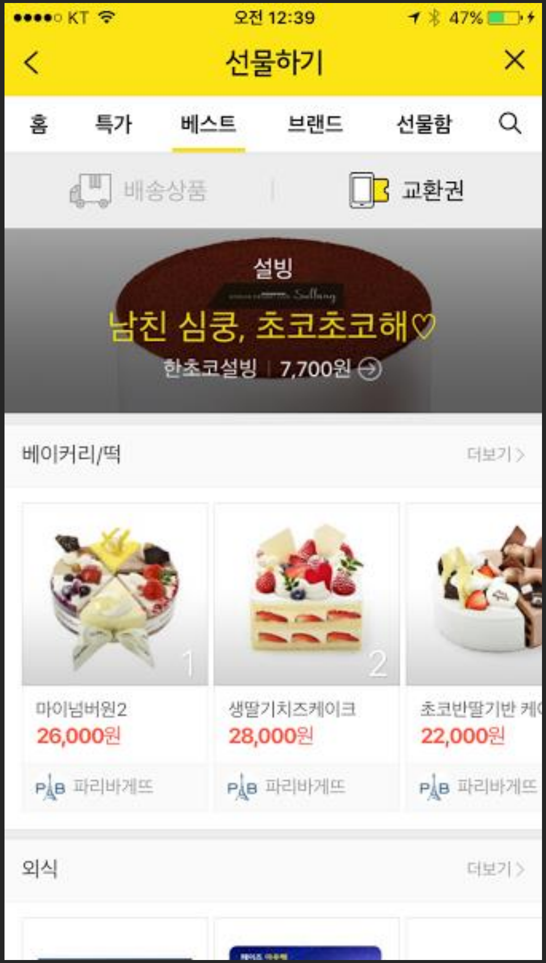
일이삼사오육칠팔구십일이삼사오육칠팔구십일이삼사오육칠…

밴딩스웨이 스커트 WMB39362 밴딩스웨이 스커트 WMB39362 밴딩스웨이 스커트 | 네이비 | 230

1 256,000원

인당 구매제한 5개 총금액 99,000,200원

Gift Best Tab



설빙

남친 심쿵, 초코초코해♡

한초코설빙 | 7,700원

베이커리/떡 더보기 >

1 마이넘버원2	2 생딸기치즈케이크	초코반딸기반 케
26,000원	28,000원	22,000원
파리바게뜨	파리바게뜨	파리바게뜨

외식 더보기 >

IU – User Info Management Server

- Programming Environments
  - OS : CentOS 7.2
  - Language : Scala
  - Framework : Finatra
  - Protocol: finagle-thrift

- Project Goal
  - CRUD and mange user information in kakao commerce(gift)
  - Gradually replace legacy Monolithic Service with Micro Service
    - Develop Micro Service using Finatra

- Project Result
  - Replace Monolithic Service Feature with Micro Service Server
  - Introduce finagle-thrift to team
  - Introduce Zookeeper to team

- Project Feature
  - Develop Micro Service Using Finatra
  - Server Cluster Management using Zookeeper
  - Fast response calling api and comfortable Integration with finagle-thrift

Hanwha Techwin

XMPP Signaling

EJABBERD & TSUNG

ejabberd & Tsung

- Programming Environments
  - OS : AWS(Ubuntu 14.04 LTS)
  - language : erlang, Scala
  - Framework : OTP, Akka
  - base open source solution – ejabberd

- Project Goal
  - Develop solution which replace old Samsung SmartCam XMPP Server solution.
  - upgrade and additional development Open Source Solution, ejabberd to meet service plan
  - Develop Load Test Program to verify Signaling Server which can control millions camera
    - upgrade and additional development Tsung – Open source Load Test Program to meet service plan
    - Develop new Load Test Program using Scala & Akka

- Project Result
  - Upgrade and additional development ejabberd to meet service plan – 30,000 TCP Connection per 1 instance(AWS c4.large)
  - Upgrade and additional development Tsung to meet service plan – 60,000 TCP Connection per 1 instance(AWS r4.large)
  - Develop new XMPP Load Test Program to meet service plan base on Akka – 20,000 TCP Connection per 1 instance(AWS c4.large)

- Project Feature
  - Occur massive TCP traffice using erlang / OPT
  - System & erlang VM Configuration to accept massive TCP traffic and obtain resilliance
  - Experience of massive traffic and handling.

ejabberd & Tsung

- Development Role
  - ejabberd – Open Source Solution
    - Can maintain 30,000 TCP Connection per instance.
    - Develop system by 20 instance can handle 60,000 TCP Connection
  - Develop and change Tsung – Open Source Load Test Program
    - Make 60,000 TCP Connection per 1 instance
    - Can occur 200 TCP Connection per second
  - Develop New Load Test Program Using Scala & Akka
    - Make 20,000 TCP Connection per 1 instance
    - Can occur 800 TCP Connection per second

Cloud VSaaS

GAIA

Gaia – Cloud VSaaS

- Programming Environments
  - OS : AWS(Ubuntu 14.04 LTS) & Azure(Ubuntu 15.10)
  - Language : Scala, Java
  - Framework : Play, Akka
- Project Goal
  - VSaaS Live Video Streaming on AWS Cloud
  - Playback Server Accept RTSP Stream and store video
- Project Result
  - Implement Auto Scaling and we can playback fluently even serviced in 3g data communication. It`s Prototype
  - Develop playback server accept RTSP Video Stream and convert to mpeg-dash and send.
- Project feature
  - Service Signaling server using ejabberd.
  - Make EC2 Cluster Concurrent Server by Akka, Play
  - WebRTC Adaptive Steaming with Kurento Media Server

Gaia – Cloud VSaaS

- Development Role
  - Design and implement Live Streaming Server(POC)
    - Design and implement RTSP to WebRTC Transcoding, live streaming server using Kurento Media Server
  - Design and implement AWS Instance Cluster(POC) Management and Business Logic Server
    - Design and implement Auto Scaling Instance management and Business Logic Server using Play Framework
  - Design and implement RTSP Endpoint server to playback RTSP Camera Video(POC)
    - Design and implement RTSP Endpoint using gStreamer
  - Design and implement Playback Server(POC)
    - Design and implement Playback Server connected with AWS S3 by change Open Source eDash-Packager

NVR/DVR Common

S – CUBE PROJECT

S – Cube Project

- Programming Environments
  - OS : Ubuntu 12.04 LTS
  - Language : C++
- Project Goal
  - Make new generation NVR Platform To substitute existing Samsung Techwin NVR Platform, Sejong
- Project Result
  - Re-Design to concrete abstract Layer in exist architecture, and export hardware specific feature by XML to implement One Source Multiple Use
- Project feature
  - Collaboration with SRIB, India(Develop in India, in the field)
  - Aim One Source, Multiple Model

S – Cube Project

- Development Role
  - Camera Manager
    - Design and implement Network Camera management and PTZ control in same network using protocol ONVIF and open source gSoap
  - Log Service
    - Design and implement module to manage, store and search all logs made by NVR using SQLite and open source Kompex Wrapper
  - Code Coverage Test Case and Management
    - Code Test code to process Test Driven Development using Gcov, CPPUnit

Soongsil University

ARM Porting

UC – II OS

uC – II OS ARM Porting

- Programming Environments
  - OS : Ubuntu 12.04 LTS
  - Language : C / ARM Assembly
  - Target Board : Odroid 7
- Project Goal
  - Port one of the RTOS operated in Window, uC – II OS to ARM Chip.
- Project Result
  - Make uC-OS II only operated in Power PC and Intel Chip to operate in ARM Architecture
  - Implement Dynamic scheduling like linux nice value. Before, it burden to engineer
- Develop Environment
  - Use s5pc110 Chip, used in Samsung Galaxy S
  - Use ARM-none-eabi Cross Compiler

μC/OS-II
The Real-Time Kernel

uC – II OS ARM Porting

- Develop Role
  - Revise U-Boot and make it to load uC – OS Kernel image in s5pc110 chip
  - Write uC – OS – II chip dependent hardware setting code by ARM Assembly to enter main entry
  - Code chip dependent driver like U-ART
  - Write interrupt code to context switch
  - Change uC-OS II Scheduler to use nice value like linux
- Performance Test
  - Using Timer inside Board to Performance Test

SAMSUNG
SOFTWARE
MEMBERSHIP

Samsung Techwin

NETWORK CAMERA VIDEO CONFERENCE

Network Camera Video Conference

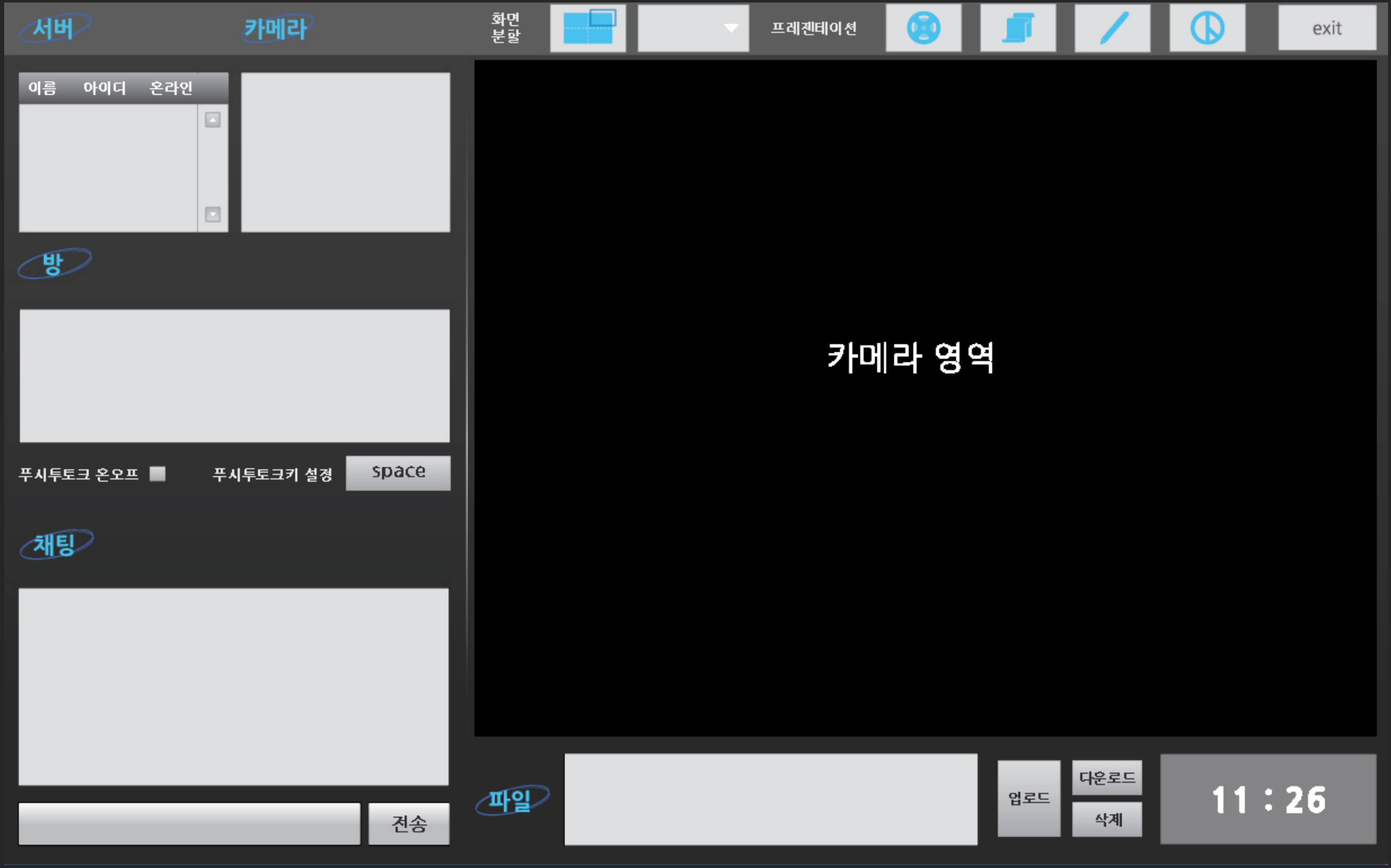
- Programming Environments
  - OS : Window 7
  - Language : C#(WPF)
- Project Goal
  - Develop Program which support existed Samsung Techwin CCTV to use Video Conference
- Project Result
  - Video Conference multiple people using CCTV and RTSP
- Project Feature
  - Re-Use Existed CCTV

Network Camera Video Conference

- Project Role
  - Program View and Business Logic
    - Design and implement Program View using WPF and All the Model Controller, Business Logic Module
  - Chatting Module
    - Design and implement to manage member, chatting room in chatting server
    - Design and implement Openfire chatting client using XMPP Library, AGS-XMPP
  - Vote Module
    - Design and implement vote module to vote specific subject in Video conference

Network Camera Video Conference

UI Design



Do Performance Yourself

DPY

DPY – Do Performance Yourself

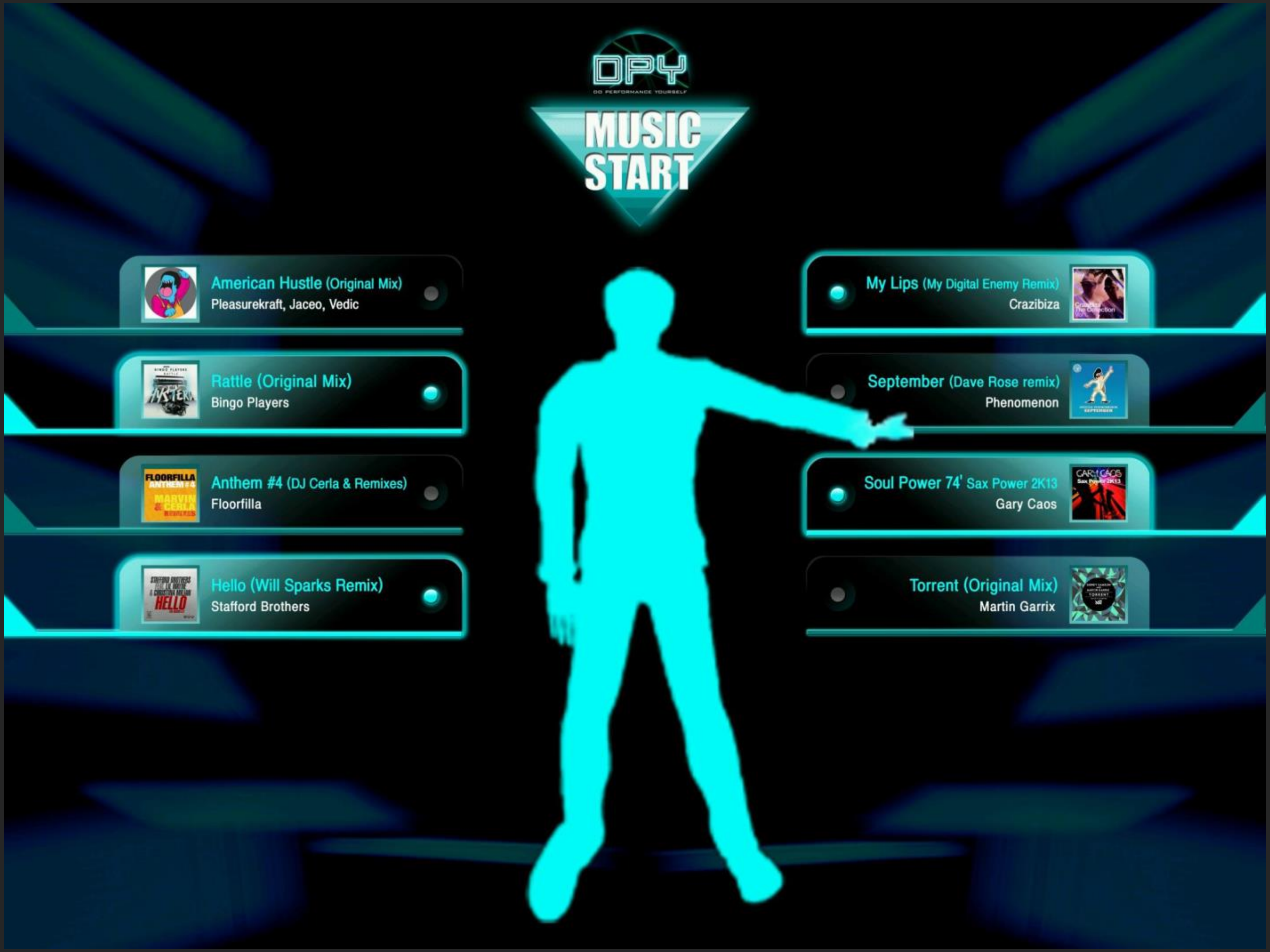
- Programming Environments
  - OS : Window 7
  - Language : C#(Unity3D)
- Project Goal
  - Develop Interaction Media Performance Content using Kinect
- Project Result
  - Develop interactive digital art React to user behavior
- Project Feature
  - User motion Capture using Machine Learning
  - Character Follow User by Unity 3D

DPY – Do Performance Yourself

- Project Role
  - Implement 3D Interactive Interface
    - Implement user UI, 3D motion, effect using Unity 3D
  - Implement module interact user and charactor
    - Implement interact module by Kinect and Zigfu
  - Implement Business Logic combine with other module
    - Combine machine Learning Module and DSP Module

DPY – Do Performance Yourself

- UI


DPY
DO PERFORMANCE YOURSELF
MUSIC START
American Hustle (Original Mix)
Pleasurekraft, Jaceo, Vedic
Rattle (Original Mix)
Bingo Players
Anthem #4 (DJ Cerla & Remixes)
Floorfilla
Hello (Will Sparks Remix)
Stafford Brothers
My Lips (My Digital Enemy Remix)
Crazibiza
September (Dave Rose remix)
Phenomenon
Soul Power 74' Sax Power 2K13
Gary Caos
Torrent (Original Mix)
Martin Garrix