

# Pankaj Bhambhani

Student at University of Massachusetts, Amherst

Phone: +14132306252  
Address: Amherst, MA - 01002  
Website: pankajb64.github.io  
Email: pbhambhani@cs.umass.edu

## \*PROFESSIONAL PROFILE

Currently pursuing my Masters at UMass Amherst, I have three years of Software Development Experience. I have worked as an SDE for Play Games24x7 Pvt. Ltd., I did my B.Tech from DA-IICT, have worked as Software Developer for KDE as a part of Google Summer of Code and Automation Engineer Intern at Oracle Retail. I have interests in Machine Learning, Natural Language Processing and Algorithms and I am looking for Data Science or Software Development internships in these areas.

## \* SKILLS

**Interest Areas** - Machine Learning, Natural Language Processing, Algorithm Design and Analysis, Software Security, Computer Networking, Information Retrieval, Artificial Intelligence

**Programming Languages** - Java, Python, C++, MySQL, JavaScript, Bash (Familiarity with: Scala, Ruby, C, HTML, CSS)

**Development Technologies** - Keras, Theano, Tensorflow, Numpy, Sk-learn, Spring, Qt, Android, Hadoop, Hbase, Ehcache, RabbitMQ, Redis, Akka, JDBC, Hibernate, MyBatis, Node.js, AngularJS, jQuery, Apache Tomcat, Django, Rails, Test-NG, JUnit, Selenium, Firebase

**Software Tools Used** - Git, SVN, Ant, Maven, Eclipse, Android Studio, Wireshark, Vim

## \* EXPERIENCE

May 2017 -  
Present

### Machine Learning for Data Science Lab Software Developer

Currently involved in the Machine Learning for Data Science (MLDS) Lab, working on BirdCast, which uses CNN based image recognition to identify bird migration patterns from radar image data. We use WSRLIB, a MATLAB library for Weather Surveillance Radar written specifically for this task, coupled with a Convolutional Neural Network trained on a VGG-16 model. Plans include scaling up the service to run on AWS clusters, and improve the performance of the CNN network.

**Technologies - MATLAB, Python.**

July 2013 - May  
2016 (2 years 10  
months)

### Play Games24x7 Private Limited Software Development Engineer

Worked as SDE for RummyCircle.com, one of the major products of Games24x7. Involved in Full Stack Web Development.

**Technologies - Java, Spring, Spring Security, Jersey (JAX-RS), Hbase, Ehcache, Terracotta, Redis, Akka, MySql, Hibernate, MyBatis, Node.js.**

Following are the key projects I worked on during my tenure.

#### 1. Analysis and Re-configuration of Fund Manager Service

Worked with the Business Intelligence (BI) team and analyzed the data (using MySQL) containing the various real-valued transactions done by the player (cash deposit, cash withdrawal, playing game for cash) as well as system (service fee, bonus release, etc.) and recorded the discrepancies in the data, most of which were concurrency issues. Traced the root of these problems to the Fund Manager Service, which handled all user and system transactions using SQL Stored Procedures. Implemented a Locking Mechanism whereby the Stored Procedures updating the balance of an entity would obtain a row-level DB lock. This significantly reduced the number of discrepancies in the transaction data.

**Technologies Used - MySQL (for analyzing the data as well as for Stored Procedures)**

**Team Size - 1**

## **2. Deposit Based Bonus**

As an extension of the existing Rummy Rewardz Store Program, where users could trade their reward points for items such as games bonuses and tournament tickets, we added a new functionality where the users could buy these items directly with cash deposits.

**Role** - Involved in the Controller and View related code changes in the Spring based application called RummyCircle (RC), which is the core user-facing service.

**Technologies Used - Java, Spring, Hibernate, JavaScript**

**Team Size - 2**

## **3. Fund Manager User Balance Bucketing**

This project involved separating the users' cash balance into two separate buckets (for marketing and legal reasons) - a deposit balance, where the user can deposit cash and use it to join games (but cannot withdraw from it) and a withdrawal buckets, which contain the users winnings and can be withdrawn any time. There were functionality changes in the all the various services using or updating a player's balance.

**Role** - Involved in code changes for RummyCircle (RC) - which displays a user's transaction history, Fund Manager(FM) - which handles the user's transactions and Player Service Portal (PSP) - an internal web application containing used by the Customer Service and Risk Mitigation Team to help users and monitor fair play activities.

**Technologies Used - Spring, Hibernate, MySQL (Stored Procedures)**

**Team Size - 3**

## **4. Event Detection System (Extension) -**

Event Detection System E was a service that captures various user events (e.g. login/logout, deposit cash, playing a game) which help directing effective marketing campaigns and improve player retention. It internally used HBase to stored all its data and analyze it. Our task was to take this service out of beta stage, and implement it throughout the various services involved in gameplay.

**Role** - Involved in code change for EDS service as well as various other gameplay services, namely RummyCircle (RC) Desktop and Mobile Service, Tournament Engine (which handles rummy games and tournaments), Secure Payment (our Payment Gateway), Fund Manager, Player Service Portal, Tournament Service Portal (used to monitor the state of running tournaments).

**Technologies used - HBase (for EDS), Java**

**Team Size - 2**

## **5. Strengthening Application Security**

Based on the feedback received from an audit carried out by a third-party on request, we implemented various defense mechanisms to improve the security of our applications against malicious users and hackers. Security features were added to prevent attacks such as Cross Site scripting (XSS), Cross Site Request Forgery (CSRF), SQL Injection, Click-jacking, Session Hacking, Session Fixation, Web Parameter Tampering among others. Also, emphasis was laid on controlling the privileges available to users of internal applications which may contain sensitive user data, using LDAP (Lightweight Data Access Protocol).

**Role** - Implemented mechanisms for defense against XSS, CSRF, SQL Injection and Session Hijacking. This involved development as well as Penetration Testing. Also was a part of access control implementation for Player Service Portal, an internal app handling player related data.

**Technologies Used - Java, Spring Security**

**Kali Linux PenTesting Tools Used - nmap, Wireshark, Burp Suite, sqlmap**

**Team Size - 3**

## **6. Scaling and Optimization of Caching Services**

The existing caching services used Ehcache client as a cache to store non-persistent data for better performance. The data was synchronized across multiple services with the help of a Terracotta server. This mechanism had low fault-tolerance, increase in user traffic often resulted in crashes which meant the entire application system would need to be restarted. We worked on fine tuning the parameters of Ehcache clients so it would cache data only as long as needed and also limit the amount of memory it would use for the same. Also, we performed a horizontal

scaling of distributed Terracotta servers so that it could easily handle a heavy load of 30,000 users (equivalent to a million requests) in peak times.

**Technologies Used - Ehcache, Terracotta, Java**

**Team Size - 2**

## 7. Optimization of Gameplay Bot

The existing gameplay bot, which was made two years before this project, could simulate gameplay for Cash Rummy Games and Rummy Tournaments by creating threads for users and games and following standard protocols (i.e. doing legit player moves). Owing to the excess use of threads (there was one thread per user per game), it could only handle a load of about 2000 users. The launch of bigger and bigger tournaments (the latest being 20k user) and the ever-increasing necessity to perform benchmark testing was the primary reason for this project. We carried out this optimization in several phases (I was involved in one of the phases). Initially, we optimized the necessary number of threads and could successfully handle upto 10k users. Later, we move to message-driven architectures like AKKA which significantly increased the user capacity of gameplay bot.

**Technologies Used, Java, AKKA**

**Team Size - 5**

## 8. Scalable REST-Based Non-Financial Service

Designed and implemented a new REST-Based service which could handle incentives that were "non-financial", i.e. other than real cash - for e.g., tickets, reward points, etc. As we were steadily moving towards a Service-Oriented Architecture, this service was designed to be stateless from the start and was scalable, achieving a benchmark of approx. a million simultaneous requests per node. The REST calls were handled using Jersey and the DB interactions were modeled using MyBatis.

**Role** - I worked on the part of the service that handles requests related to tournament tickets.

**Technologies Used - Jersey (JAX-RS), MyBatis**

**Team Size - 2**

*May 2012 -  
August 2012 (4  
months)*

## KDE (Google Summer of Code)

Software Developer

Integrated features of Facebook into KDE's micro blogging client, Choqok (Under the program **Google Summer of Code 2012**). Using KDE's **libkfbapi**, on which I worked the previous year, we added the ability to view posts, like and comment on them and also create a new post.

Technology used: **KDE4 based on Qt** (C++ based GUI Framework).

Link - <https://github.com/pankajb64/choqok-facebook>

*May 2011 - July  
2011 (3 months)*

## KDE

Software Developer

Build base KDE library, namely KFBAPI, which formed the ground for Integrating Facebook Features for Choqok (which was done next summer under Google Summer of Code 2012).

Technology used: **KDE4 based on Qt** (C++ based GUI Framework).

*December 2012 -  
May 2013 (6  
months)*

## Oracle

Interim Automation Engineer

Worked as an Automation Engineer with Merchandise Operations Management (MOM) Team at Oracle Retail on products like RMS (Retail Merchandising System) and Allocation. Used technologies like **QTP (Quick Test Professional)** for RMS and **Selenium Webdriver** for scripting and execution of test cases.

## \* OTHER PROJECTS

*June 2017*

In the Bin! (Google Hacking For Good 2017)

<https://github.com/HackingForGood/HackU>

In the bin is a location-based crowd-sourced web application where users can submit locations of publicly-accessible trash and recycle bins near by. Built as part of the Google **Hacking For Good** hackathon at Harvard Business Labs on June 17, 2017.

View the app at <http://bit.ly/inthebinapp>

**Technologies - Firebase Web, Google Maps JavaScript API, Material Design Lite**

*October 2016*

## Pictionary with Jibo

<https://github.com/pankajb64/jibo-pictionary>

Taught the robot Jibo to play Pictionary on a set of hand-drawn images. We used Jibo SDK to capture a picture and send it off to be classified by our custom-trained tensor flow neural networks. We used openCV to detect the playing area within the board and crop off the rest of the image and make it black and white, we then passed it off to our TensorFlow model to be classified. We built the neural network using hundreds of images pulled automatically from Bing Image Search and used TensorFlow's Image Recognition Algorithm "Inception" to train the model.

**Technologies: TensorFlow, Jibo SDK, OpenCV, Python**

*April 2015 to May 2015*

## Duckworth Lewis (D/L) Calculator for Android

<https://github.com/pankajb64/duckworth-lewis-d-l-calculator-android>

Designed and Built Duckworth Lewis Calculator - an Android application which lets users predict scores and results in rain interrupted cricket matches.

**Technology: Android.**

Link to play store app - <https://play.google.com/store/apps/details?id=com.fl.dlc>

*January 2015 - February 2015*

## Komey

<https://github.com/pankajb64/komey>

Komey is a linguistic bot capable of automating build tasks by reading textual instructions and performing suitable actions.

### \* ACADEMIC PROJECTS

*May 2017 - Present*

## Evaluating Machine Learning Techniques for Website Fingerprinting Attacks

<https://github.com/pankajb64/webfp-crawler-phantomjs/>

Currently pursuing an independent study which evaluates the assumptions made by different machine learning models for targeting Website Fingerprinting (WF) attacks on the Tor network. Current learning models include Naïve Bayes, Decision Trees and SVM, but they all employ certain restrictions (e.g., user visits only a small set of web pages, page content doesn't change over time) that aren't realistically attainable. We evaluate the assumptions made by such models and the performance in their absence. Plans include evaluating deep learning based models for such attacks, and propose adversarial examples for the same.

**Technologies - Numpy, Scikit-learn, Python**

*October 2016 to December 2016*

## Image Captioning with Visual Attention

[https://github.com/pankajb64/image\\_caption\\_using\\_attention](https://github.com/pankajb64/image_caption_using_attention)

We used gradient descent based training algorithms incorporated with an attention mechanism to build a deep learning model that learns to generate image descriptions given an unseen image, inspired by the state-of-the-art work in this field. Our results showed that attention based deep learning models can learn to produce good image captions and can also generate visualizations to infer how a model can focus on certain parts of the image while predicting the next word. We used a Convolutional Neural Network (CNN) to generate attention based features and a Long Short-term Memory Network (LSTM) as a recurrent net for generating image captions.

**Technologies: Keras, Theano, Python**

October 2016 to  
December 2016

## Content Summarization of a Book using its Index

<https://github.com/pankajb64/bookesis>

Generating coherent summaries of long documents such as books is a relatively untouched field, with most of the research work being focused on short documents. In this Project, we looked at summarization of books which have a back-index and see if it can provide some useful information in generating good summaries. We also provided a new dataset for the evaluation of such summarization systems.

**Technologies: NLTK, Python**

October 2012 to  
November 2012

## Certified Electronic Mail

<http://slidesha.re/certemail>

Studied the properties and commented on possible (desirable + implementable) combinations for a Certified Electronic Mail System for providing Security to Electronic Mail with a focus on improving security features of existing widely used security protocols: S-Mime, PGP. Team Size - 5

Link to Presentation - <http://slidesha.re/certemail>

October 2012 to  
November 2012

## Synthesize Reactions using Chemical Computing Constructs

<https://github.com/pankajb64/reaction-synthesizer>

Analyzed how various computing constructs could be synthesized using appropriate chemical reactions and wrote a model class for the same..

Link to project source - <https://github.com/pankajb64/reaction-synthesizer>

**Technology - Java.** Team Size - 3

January 2012 to  
April 2012

## Ecodena

<https://github.com/pankajb64/ecodena>

Created an Online Judge System, which attempts to motivate novice users towards problem solving by providing comprehensive error reporting and a variety of tutorials.

**Technology: Django** (based on Python). Team Size - 9

Link to project source - <https://github.com/pankajb64/ecodena>

### \* EDUCATION

September 2016  
- Present

## University of Massachusetts Amherst

Master of Science in Computer Science

GPA - 4.0

Courses Taken : **Machine Learning, Natural Language Processing**

2009 - 2013

## DA-IICT

Bachelor of Technology (B.Tech.), Information and Communication Technology,

CPI 8.6/10

Relevant Courses Taken - **Algorithms, Data Structures, Computer Networks, Databases, Operating Systems, Systems Software, Cryptography, Security Protocols, Graph Theory, Computer Organisation, Natural Computing**