A PROJECT REPORT

on

DevDest Programmers' Application

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
INFORMATION TECHNOLOGY

BY

DEVANSH KASHYAP	1706036
HARSH SINGH	1706038
HIMANSHU	1706040

UNDER THE GUIDANCE OF
PROF. DIVYA KUMARI

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA - 751024

May 2020

A PROJECT REPORT

on

DevDest Programmers' Application

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN INFORMATION TECHNOLOGY

BY

DEVANSH KASHYAP	1706036
HARSH SINGH	1706038
HIMANSHU	1706040

UNDER THE GUIDANCE OF

PROF. DIVYA KUMARI

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA -751024

May 2020

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certify that the project entitled

DevDest Programmers' Application

submitted by

Devansh Kashyap	1706036
Harsh Singh	1706038
Himanshu	1706040

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2019-2020, under our guidance.

Date: / /

(Prof. Guide Name)
Project Guide

Acknowledgements

We are profoundly grateful to Prof. Divya Kumari for her expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

Devansh Kashyap
Harsh Singh
Himanshu

ABSTRACT

DevDest is a developer/programmer questionnaire application for an android platform where developers across the world can ask questions related to technical stuff. A common place for all the developers is a great initiative and DevDest is designed for the same. Also, with the growing need of personal customization and to save time in searching and answering questions, we have provided a full-time machine learning support to the input and output queries, which in accordance with the queries classifies them into different tags.

This application also has a chat bot made using NLP as a help and support bot for the basic queries of the users of the application. The chat bot is web and telegram integrated which can handle basic queries on the working of the application and can handle small talks too to reply like a friend to the user.

Contents

List of Figures

Chapter 1

Introduction

4.1 Purpose

This project provides an overview of how our application works. The plan will include, but is not restricted to, a summary of the system functionality, the scope of the project from the perspective of the "DevDest Application" team, various estimates, project risks and how those risks will be mitigated, the process by which we developed the project, and metrics and measurements that were recorded throughout the project.
This document is written primarily for the stakeholders, such as the developers, programmers and maintainers of the system.

4.2 Goals of Project

This project is developed for the coder family around the globe, and various people who are experts in their domains of programming. The goals of this project are:

- Users should be able to sign up with their existing email accounts.
- They should be able to make their own profile with their own fields of interests, their age group, proficient languages etc.
- Users should be able to post answers to new problems, as well as post new problems if any into the main question bank.
- Experts should get questions filtered on their choice of interests only, hence reducing the spams and increasing accuracy.
- Users should be able to access the answers available on the application, and also search for them.

Chapter 2

Literature Survey

DevDest application has been developed in accordance with the Stack Exchange Network's StackOverflow website. Stack Overflow is a question and answer site for professional and enthusiast programmers. It is by programmers, for programmers, with the ultimate intent of collectively increasing the sum total of good programming knowledge in the world. No dependency on the programming language or the operating system. It's better than other Q&A sites, because you don't have to read through a lot of discussion to find the right answer, if it's in there somewhere.

DevDest also deals with questions and answers in a similar way. Using machine Learning algorithms to predict the tags and the classifying different questions based on those tags is its main job. However, it is also loaded with other features such as a chat bot for interaction and a feed page.
DevDest uses the following technologies and favours from the internet:

1. Android Studio
2. Anaconda
3. Python 3. 7
4. Dialogflow
5. Various machine learning concepts including One vs rest classifier

Throughout the development of the various machine learning algorithms, much research has already been done and DevDest did not require any such research to be done further. However, for choosing the most valuable features and classifying algorithm, ample amount of research on terms of accuracy and efficiency has been done for this project and the result is an optimised application.

References from various online literature such as *Jeff Atwood (16 April 2008). "Introducing Stackoverflow.com". Coding Horror; Stack Overflow Website - https://stackoverflow.com; Geeks for Geeks; Android Developers;* etc. has been taken as required.

Gonzalez, Jose & Flores, Juan & Graff, Mario & Calderon, Felix. (2015) did research on Multi-class multi-tag classifier system for StackOverflow questions. They have used a 5-way multiclass classifier system. The results obtained by this classification scheme are discussed, by analysing certain score metrics of the classifier system.

Paul Heymann, Daniel Ramage, and Hector Garcia-Molina have predicted tags based on page text, anchor text, surrounding hosts, and other tags applied to the standards. We have gained idea about the classification.

AUTOTAGGING MUSIC USING SUPERVISED MACHINE LEARNING takes as input acoustic features and predicts social tags mined from the web. We have tested our requirements based on this scenario and prepared for the application.

Chapter 3

Software Requirements Specification

3.1 External Interfaces

There are not many software, hardware and other system requirements that the user must ensure to achieve a fluidic access to all features of the application. These are listed out as follows.

3.1.1 User Interfaces

This project is an application with a GUI (Graphical User Interface), hosted on Android platform, such that the user will easily and instinctively be able to interact with the interface to get the desired functionality, as one of the chief purposes for this project is to improve user convenience.

3.1.2 Hardware Interfaces

User of this application is any Android device user that loads this application to their device. All of the users are in the same class, only one type of user exists. Operating environment is, as just mentioned above, is an Android OS mobile device. An android device that can support basic dependencies of the application is expected for proper user experience.

3.1.3 Software Interfaces

The application will be loaded with the models and algorithm which will perform the task and the functionalities. The backend is dedicated to machine learning processing, database management and transfer of data in various phases.

3.1.4 Communications Interfaces

A moderate speed Internet Connectivity is required for the proper experience of the application.

3.1.5 Memory Constraints

This first version of this application takes about 30MB of storage space in the mobile phones resulting to be very space optimised application. The size may vary in successive versions depending on future patches and updates.

3.1.6 Operations

- All the user data is backed up on the cloud in our database and recommendations are made accordingly. This data improves tag recognition for the particular user and also in finding the appropriate solution to his/her queries.
- The posted questions and queries are available to the user at any time sooner or later in the database as per his choice. Also, no copy of the data is available for anyone except the original user. Hence, data is safe from intrusion.

3.1.7 Site Adaptation Requirements

- To verify that the user is a not a spammer and to provide access to genuine users, a verified email account is required to initiate signup in the application.
- A wired or wireless Internet connection would be needed for the application to function and communicate with the various users available on the application.

3.2 Functional Requirements

The following will be the major functional requirements from the project, apart from which many minor functionalities might also be introduced either from the beginning or in future versions.

3.2.1 User Login & Sign Up

- Login and password: This is the criterion through which the authenticity of a client is checked. The system initially prompts the user to fill their login ID and respective password, and then proceeds further.
- New User Creation: In case the user has not registered for an ID, he/she can check the Sign-Up option and will be redirected to a new page to create a new account.

3.2.2 Input of Queries and Details

- Post Question: User will post a question as an input, providing details of the problem or output that they want to be executed. For instance, the user might post a question like "How to find sum of all elements in an array?".
- Search Questions: User should enter the question on the search tab to find the solutions already given by other users earlier before it .

3.2.3 Search a Question

- Some questions might already be present in the database and the user could save time by directly searching for the problem here. This function allows the user to get solutions related to his query quickly.

3.2.4 View Account Details

In case the user wants to alter or view his/her account details, this function is available. For instance, changing the account password, viewing the recent queries posted etc.

3.3 Use Case

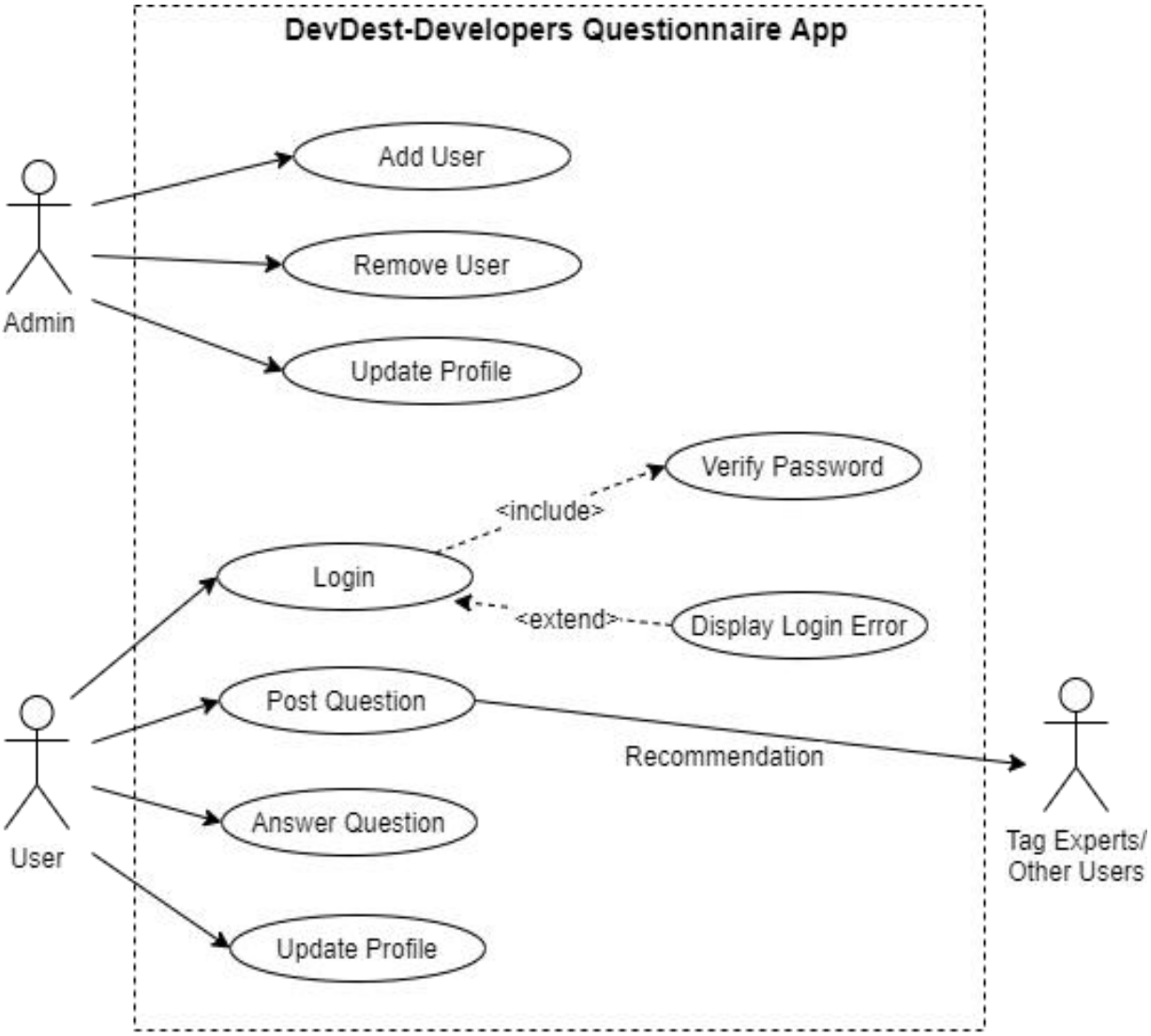


Figure 1: Use Case Diagram

3.4 Non-Functional Requirements

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. The requirements are stated in the following sections in measurable terms.

3.4.1 Performance

- Speed of the Internet connection from the remote computer and the speed of the connection to the server will affect performance. The DevDest Programmers' application will be designed to be capable of operating even with slow Internet connections e.g. a standard 56K wireless connection to the network.
- The DevDest Application will be designed to operate on a standard Android smartphone. Response times to commands should be minimal.

3.4.2 Reliability

The system has to be highly reliable due to the importance of data. The system will run 7 days a week, 24 hours a day. Also, since it is a recommendation driven product, the system has to be reliable such that all tags and questions posted by the user are received by the appropriate accurately, and are executed properly; incorrect questions should not be sent by the application. Moreover, since the software improves over time after learning from the habits of the user, it needs to be reliable in the recommendations it gives to the user based on what it has learned about the preferences of the user.

3.4.3 Availability

- Since the software is being developed for the developers' doubts and convenience in getting solutions to problems, it is a huge value service for the consumer, and so consumer convenience is key. To ensure consumer convenience, the system has to be available 100% for the user and is used 24 hours a day, 7 days a week and 365 days a year.
- Availability of an internet connection needs to be ensured for proper functioning.

3.4.4 Security

There are not many security risks associated with using the DevDest application as it is designed to operate in a secured way, including a little amount of data exchanges over the internet. When accessing the application, the user needs to be assured that intruders, such as hacker attempts and third-party invasions, cannot have access to their accounts. Thus, proper authentication at the time of signup is essential. The system will accept this information as proof of the identity and allow the user to access the application. Also, the data of the user, if saved, should be safely encrypted and protected against any kinds of security threats like hacks or other cyber or socio-engineering attacks.

3.4.5 Maintainability

The rapid progression of the technology means frequent software updates from manufacturers. These updates come in the form of application updates. Generally speaking, the app updates can be handled automatically by the user's smartphone or tablet and don't require much user attention.

3.4.6 Portability

Changes must be verified once per day at least. The system should automatically provide notification to users by email about posts overdue, query results, new questions etc. Also, the software should be re-installable and accessible from all smartphones, given that the user is able to authenticate his/her login on any such device. So none of the code should be host-dependent.

Chapter 4

Requirement Analysis

User classes and their roles:

A user and his/her roles give us a clear view of the requirement and types of job the user is going to perform using the Fabula application.

1. **End Users:** The end users for DevDest application are expected to have some technical background and prior technical knowledge. The user interaction is easy to use and a friendly environment is provided.
   Posting new questions or answering the previously uploaded questions would be the main job of end users.
2. **Developers:** Developers will be the primary maintainers of the application and may have to perform some system related tasks. They need to have expertise in technical fields such as Android development, Data analysis, Software engineering, use of statistical tools and advance knowledge to improvise the application.
   The role of these developers as primary maintainers of the application will include the duties like improvising the application on the basis of performance, security, risks and safety. The role also dictates to make considerable progressions to meet all the future goals for this application.
3. **Database Administrator (DBA):** The person who has central control of the whole database system is called Database administrator or DBA. The DBA coordinates all the activities of the database system. They require technical expertise in database management. The DBA does not directly interact with the application.
   The role of DBA in this application is to define and control the access methods for the different users, alter the physical organization to improve performance, responsible for periodically backing up the database, monitor the jobs running on the database and ensure that the performance is not degraded by very expensive tasks submitted by some users.

Chapter 5

System Design

5.1 Data Flow Diagram

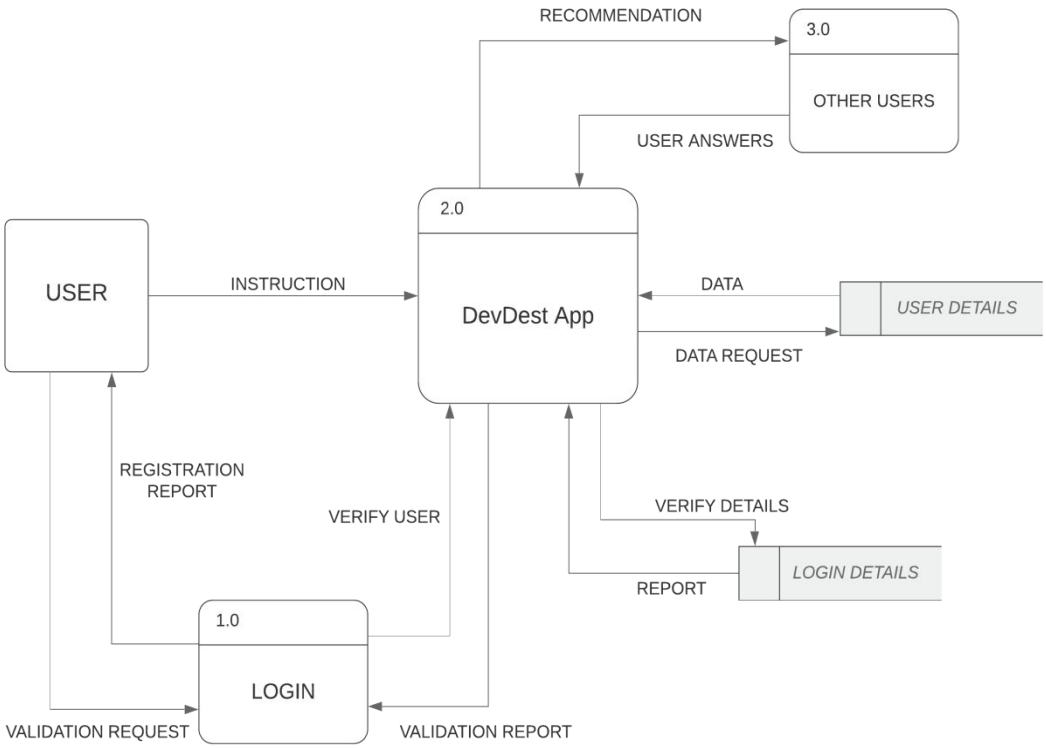


Figure 2: Data Flow Diagram (Level 1)

5.2 Activity Diagram

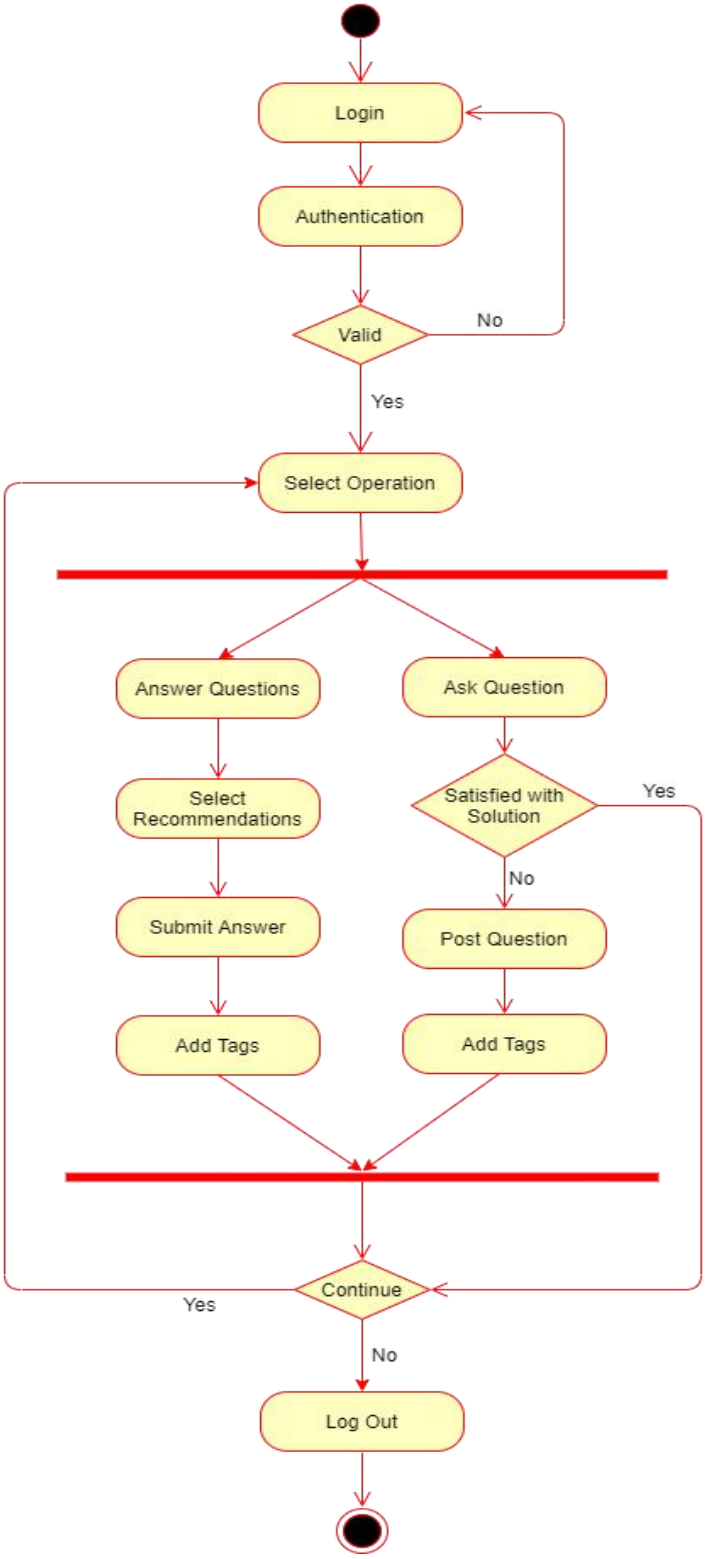


Figure 3: Activity Diagram

5.3 Sequence Diagram

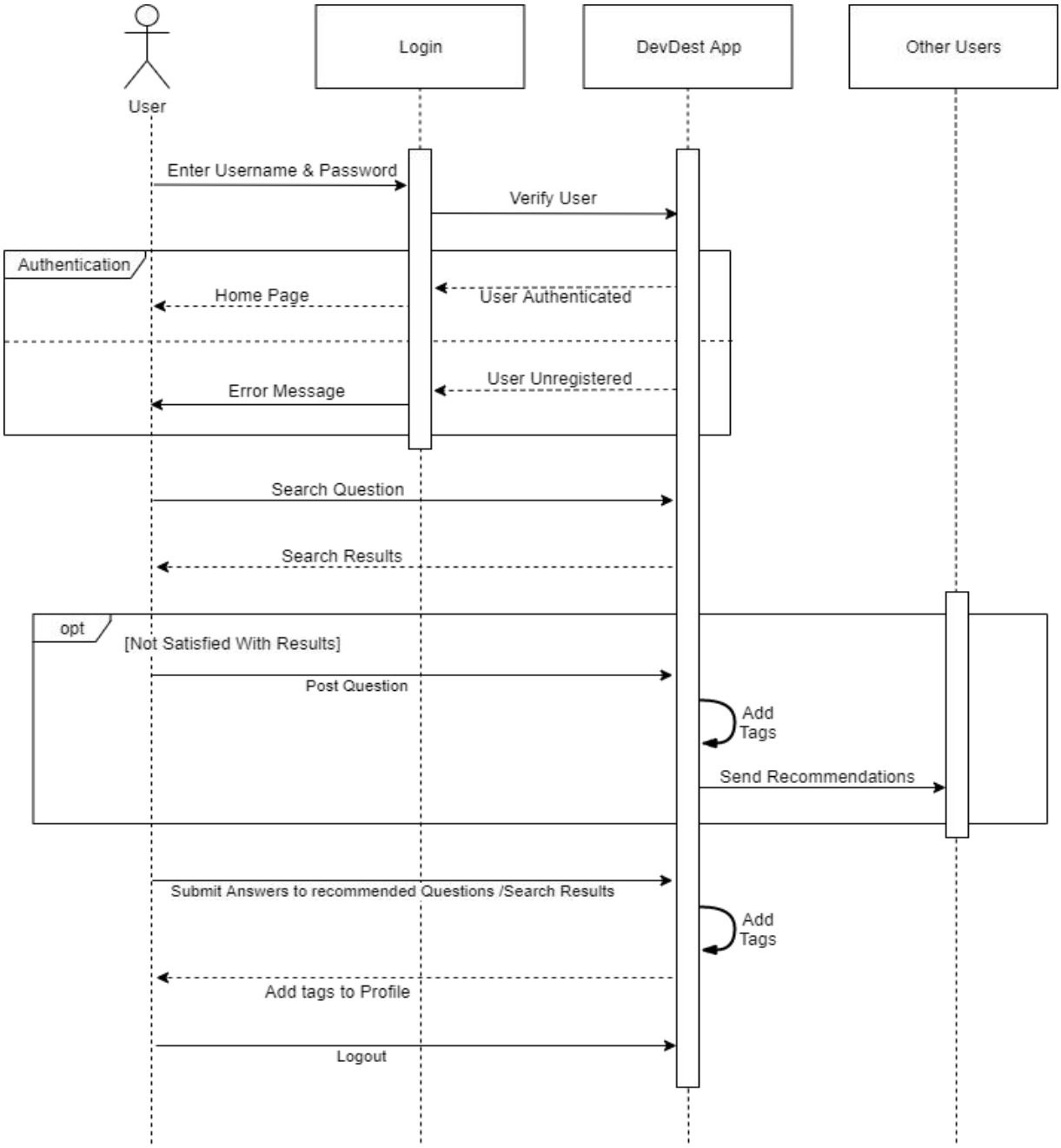


Figure 4: Sequence Diagram

Chapter 6

System Testing

WRITE HERE.

6.1 Test Cases and Test Results

Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	Tag Predictor test	Question as text	Error	0.06 RMSE
T02	Android App test	Navigation and features test	Working	Correct Responce
T03	Chatbot testing	Intents Input	Got right response	Responsiveness

Note: Testing should be performed manually

Chapter 7

Project Planning

Activity Planning Table

Activity	Start Week	No. of Weeks
Project ideation and Literature Review	3rd week of January	3
Requirment Analysis	1st week of February	1
Data Gathering and EDA	2nd week of February	2
System Design	3rd week of February	1
Recommendation Model Building	1st week of March	2
Chatbot Building	2nd week of March	1
Andriod Application Building	3rd week of March	3
System Deployment	1st week of April	1
System Integreation	2nd week of April	1
System Testing	2nd week of April	1
Documentation of Project Report	3rd week of April	1

Chapter 8

Implementation

DevDest application is made on android studio, and features of the application includes feature of tag prediction and and a chatbot too. The chatbot is made using Dialogflow and the tag prediction system is made using Python 3.7 on jupyter notebook using machine learning libraries. The code for the tag prediction is shown below:

In [1]:

```
import pandas as pd
import numpy as np


import matplotlib.pyplot as plt
import matplotlib.lines as mlines
import seaborn as sns

import warnings

import pickle
import time

import re
from bs4 import BeautifulSoup
import nltk
from nltk.tokenize import ToktokTokenizer
from nltk.stem.wordnet import WordNetLemmatizer
from nltk.corpus import stopwords
from string import punctuation

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import LatentDirichletAllocation
from sklearn.preprocessing import MultiLabelBinarizer
from sklearn.model_selection import train_test_split
from sklearn.model_selection import learning_curve
from sklearn.model_selection import ShuffleSplit
from sklearn.dummy import DummyClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import SGDClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.multiclass import OneVsRestClassifier
from sklearn.svm import LinearSVC
from sklearn.linear_model import Perceptron
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn import model_selection
from sklearn.metrics import make_scorer
from sklearn.metrics import confusion_matrix
from sklearn.metrics import hamming_loss
from sklearn.cluster import KMeans


import logging

from scipy.sparse import hstack

warnings.filterwarnings("ignore")
plt.style.use('bmh')
%matplotlib inline
```

In [2]:
```
ques = pd.read_csv("Questions.csv", encoding ='ISO-8859-1')
```

In [3]:
```
tags = pd.read_csv("Tags.csv", encoding="ISO-8859-1")
```

In [4]:
```
# Questions.csv
```

In [5]:
```
dcols = ['OwnerUserId', 'CreationDate', 'ClosedDate']
ques.drop(dcols, axis = 1, inplace = True)
ques.head()
```

Out[5]:

	Id	Score	Title	Body
0	80	26	SQLStatement.execute() - multiple queries in o...	<p>I've written a database generation script i...
1	90	144	Good branching and merging tutorials for Torto...	<p>Are there any really good tutorials explain...
2	120	21	ASP.NET Site Maps	<p>Has anyone got experience creating ...
3	180	53	Function for creating color wheels	<p>This is something I've pseudo-solved many t...
4	260	49	Adding scripting functionality to .NET applica...	<p>I have a little game written in C#. It uses...

In [6]:
```
tags['Tag'] = tags['Tag'].astype(str)
```

In [7]:
```
grouped_tags = tags.groupby("Id")['Tag'].apply(lambda tags: ' '.join(tags))
grouped_tags.reset_index()
g_final = pd.DataFrame({'Id':grouped_tags.index, 'Tags':grouped_tags.values})
```

In [8]:
```
df = ques.merge(g_final, on='Id')
```

In [9]:
```
df.head(5)
```

Out[9]:

	Id	Score	Title	Body	Tags
0	80	26	SQLStatement.execute() - multiple queries in o...	<p>I've written a database generation script i...	flex actionscript-3 air
1	90	144	Good branching and merging tutorials for Torto...	<p>Are there any really good tutorials explain...	svn tortoisesvn branch branching-and-merging
2	120	21	ASP.NET Site Maps	<p>Has anyone got experience creating ...	sql asp.net sitemap
3	180	53	Function for creating color wheels	<p>This is something I've pseudo-solved many t...	algorithm language-agnostic colors color-space
4	260	49	Adding scripting functionality to .NET applica...	<p>I have a little game written in C#. It uses...	c# .net scripting compiler-construction

In [10]:
```
df.dropna(axis = 1, how ='all', inplace = True)
df.dropna(axis = 0, how ='all', inplace = True)
```

In [11]:
```
df.drop_duplicates(keep = 'first', inplace = True)
```

In [12]:
```
d_null = [col for col in df.columns if df[col].isnull().sum() > (0.78*df.shape[0])]
df.drop(d_null, axis = 1, inplace = True)
df.head(5)
```

Out[12]:

	Id	Score	Title	Body	Tags
0	80	26	SQLStatement.execute() - multiple queries in o...	<p>I've written a database generation script i...	flex actionscript-3 air
1	90	144	Good branching and merging tutorials for Torto...	<p>Are there any really good tutorials explain...	svn tortoisesvn branch branching-and-merging
2	120	21	ASP.NET Site Maps	<p>Has anyone got experience creating ...	sql asp.net sitemap
3	180	53	Function for creating color wheels	<p>This is something I've pseudo-solved many t...	algorithm language-agnostic colors color-space
4	260	49	Adding scripting functionality to .NET applica...	<p>I have a little game written in C#. It uses...	c# .net scripting compiler-construction

In [13]:
```
df = df[df['Score'] > 4]
df.head(5)
```

Out[13]:

	Id	Score	Title	Body	Tags
0	80	26	SQLStatement.execute() - multiple queries in o...	<p>I've written a database generation script i...	flex actionscript-3 air
1	90	144	Good branching and merging tutorials for Torto...	<p>Are there any really good tutorials explain...	svn tortoisesvn branch branching-and-merging
2	120	21	ASP.NET Site Maps	<p>Has anyone got experience creating ...	sql asp.net sitemap
3	180	53	Function for creating color wheels	<p>This is something I've pseudo-solved many t...	algorithm language-agnostic colors color-space
4	260	49	Adding scripting functionality to .NET applica...	<p>I have a little game written in C#. It uses...	c# .net scripting compiler-construction

In [14]:
```
df.drop_duplicates(inplace = True)
```

In [15]:
```
df.drop(columns=['Id', 'Score'], inplace=True)
df.head(5)
```

Out[15]:

	Title	Body	Tags
0	SQLStatement.execute() - multiple queries in o...	<p>I've written a database generation script i...	flex actionscript-3 air
1	Good branching and merging tutorials for Torto...	<p>Are there any really good tutorials explain...	svn tortoisesvn branch branching-and-merging
2	ASP.NET Site Maps	<p>Has anyone got experience creating ...	sql asp.net sitemap
3	Function for creating color wheels	<p>This is something I've pseudo-solved many t...	algorithm language-agnostic colors color-space
4	Adding scripting functionality to .NET applica...	<p>I have a little game written in C#. It uses...	c# .net scripting compiler-construction

In [16]:
```
df['Tags'] = df['Tags'].apply(lambda x: x.split())
```

In [17]:
```
all_tags = [i for s in df['Tags'].values for i in s]
```

In [18]:
```
len(all_tags)
```

Out[18]: 286401

In [19]:
```
my_set = set(all_tags)
unique_tags = list(my_set)
len(unique_tags)
```

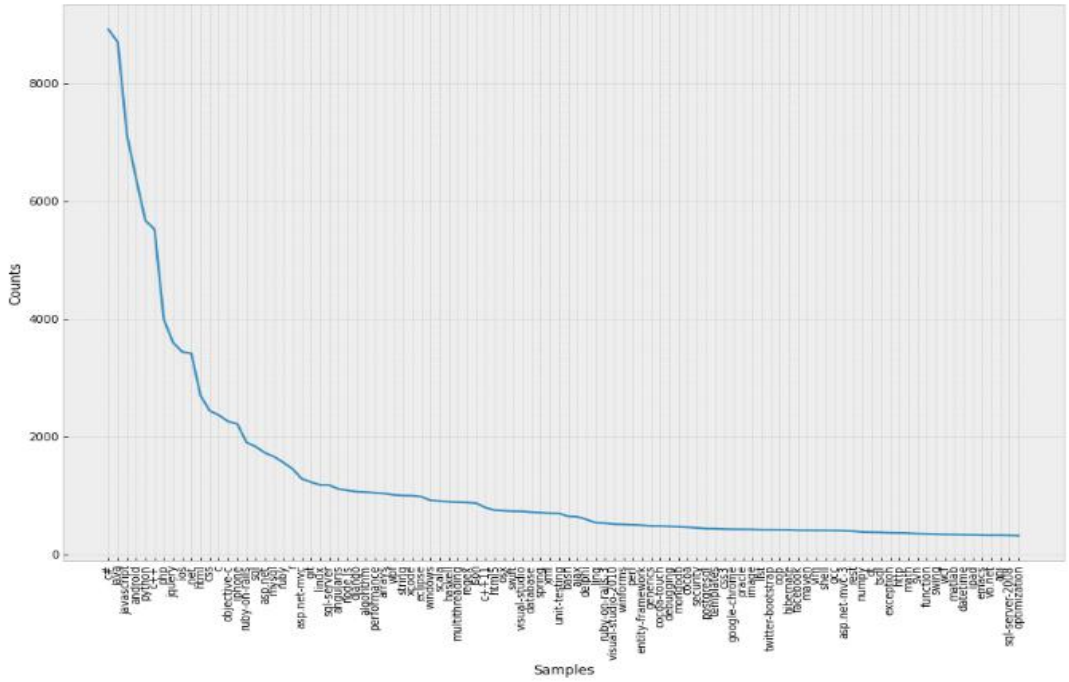
Out[19]: 16632

In [20]:
```
keywords = nltk.FreqDist(all_tags)

keywords = nltk.FreqDist(keywords)

frequencies_words = keywords.most_common(100)
tags_features = [word[0] for word in frequencies_words]
```

In [22]:
```
fig, ax = plt.subplots(figsize=(15, 10))
keywords.plot(100, cumulative=False)
```



Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x24bccfa24c8>

In [23]:
```
def most_common(tags):
    tags_filtered = []
    for i in range(0, len(tags)):
        if tags[i] in tags_features:
            tags_filtered.append(tags[i])
    return tags_filtered

df['Tags'] = df['Tags'].apply(lambda x: most_common(x))
df['Tags'] = df['Tags'].apply(lambda x: x if len(x)>0 else None)
```

In [24]:
```
df.dropna(subset=['Tags'], inplace=True)
```

In [25]:
```
# Text processing for the body of text
```

In [26]:
```
# Converting html to text in the body

df['Body'] = df['Body'].apply(lambda x: BeautifulSoup(x).get_text())
```

In [27]:
```
def clean_text(text):
    text = text.lower()
    text = re.sub(r"what's", "what is ", text)
    text = re.sub(r"\'s", " ", text)
    text = re.sub(r"\'ve", " have ", text)
    text = re.sub(r"can't", "can not ", text)
    text = re.sub(r"n't", " not ", text)
    text = re.sub(r"i'm", "i am ", text)
    text = re.sub(r"\'re", " are ", text)
    text = re.sub(r"\'d", " would ", text)
    text = re.sub(r"\'ll", " will ", text)
    text = re.sub(r"\'scuse", " excuse ", text)
    text = re.sub(r"\'\n", " ", text)
    text = re.sub(r"\'\xa0", " ", text)
    text = re.sub('\s+', ' ', text)
    text = text.strip(' ')
    return text
df['Body'] = df['Body'].apply(lambda x: clean_text(x))
```

In [28]:
```
token=ToktokTokenizer()
punctuation
```

Out[28]: '!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'

In [29]:
```
punct = '!"#$%&\'()*+,./:;<=>?@[\\]^_`{|}~'
```

In [30]:
```
def strip_list_noempty(mylist):
    newlist = (item.strip() if hasattr(item, 'strip') else item for item in mylist)
    return [item for item in newlist if item != '']

def clean_punct(text):
    words=token.tokenize(text)
    punctuation_filtered = []
    regex = re.compile('[%s]' % re.escape(punct))
    remove_punctuation = str.maketrans(' ', ' ', punct)
    for w in words:
        if w in tags_features:
            punctuation_filtered.append(w)
        else:
            punctuation_filtered.append(regex.sub('', w))

    filtered_list = strip_list_noempty(punctuation_filtered)

    return ' '.join(map(str, filtered_list))

df['Body'] = df['Body'].apply(lambda x: clean_punct(x))
```

In [31]:
```
df['Body'][5]
```

Out[31]:
```
'i am working on a collection of classes used for video playback and recording i have one main class which acts like the pu
blic interface with methods like play stop pause record etc then i have workhorse classes which do the video decoding and v
ideo encoding i just learned about the existence of nested classes in c++ and i am curious to know what programmers think a
bout using them i am a little wary and not really sure what the benefitsdrawbacks are but they seem according to the book i
am reading to be used in cases such as mine the book suggests that in a scenario like mine a good solution would be to nest
the workhorse classes inside the interface class so there are no separate files for classes the client is not meant to use
and to avoid any possible naming conflicts i do not know about these justifications nested classes are a new concept to me
just want to see what programmers think about the issue'
```

In [32]:
```
lemma = WordNetLemmatizer()
stop_words = set(stopwords.words("english"))

def lemitizeWords(text):
    words=token.tokenize(text)
    listLemma=[]
    for w in words:
        x=lemma.lemmatize(w, pos="v")
        listLemma.append(x)
    return ' '.join(map(str, listLemma))

def stopWordsRemove(text):

    stop_words = set(stopwords.words("english"))

    words=token.tokenize(text)

    filtered = [w for w in words if not w in stop_words]

    return ' '.join(map(str, filtered))
df['Body'] = df['Body'].apply(lambda x: lemitizeWords(x))
df['Body'] = df['Body'].apply(lambda x: stopWordsRemove(x))
```

In [33]:
```
df['Title'] = df['Title'].apply(lambda x: str(x))
df['Title'] = df['Title'].apply(lambda x: clean_text(x))
df['Title'] = df['Title'].apply(lambda x: clean_punct(x))
df['Title'] = df['Title'].apply(lambda x: lemitizeWords(x))
df['Title'] = df['Title'].apply(lambda x: stopWordsRemove(x))
```

In [34]:
```
df.head(5)
```

Out[34]:

	Title	Body	Tags
1	good branch merge tutorials tortoisesvn	really good tutorials explain branch merge apa...	[svn]
2	asp.net site map	anyone get experience create sql-based asp.net...	[sql, asp.net]
3	function create color wheel	something pseudo-solved many time never quite ...	[algorithm]
4	add script functionality .net applications	little game write c use database back-end trad...	[c#, .net]
5	use nest class case	work collection class use video playback recor...	[c++, oop]

In [35]:
```
from sklearn.preprocessing import MultiLabelBinarizer
X1 = df['Body']
X2 = df['Title']
y = df['Tags']
multilabel_binarizer = MultiLabelBinarizer()
y_bin = multilabel_binarizer.fit_transform(y)
```

In [36]:
```
vectorizer_X1 = TfidfVectorizer(analyzer = 'word',
                                min_df=0.0,
                                max_df = 1.0,
                                strip_accents = None,
                                encoding = 'utf-8',
                                preprocessor=None,
                                token_pattern=r"(?u)\S\S+",
                                max_features=1000)

vectorizer_X2 = TfidfVectorizer(analyzer = 'word',
                                min_df=0.0,
                                max_df = 1.0,
                                strip_accents = None,
                                encoding = 'utf-8',
                                preprocessor=None,
                                token_pattern=r"(?u)\S\S+",
                                max_features=1000)
```

In [37]:
```
X1_tfidf = vectorizer_X1.fit_transform(X1)
X2_tfidf = vectorizer_X2.fit_transform(X2)
```

In [38]:
```
X_tfidf = hstack([X1_tfidf,X2_tfidf])
X_train, X_test, y_train, y_test = train_test_split(X_tfidf, y_bin, test_size = 0.2, random_state = 0)
```

In [39]:
```
def avg_jacard(y_true,y_pred):

    jacard = np.minimum(y_true,y_pred).sum(axis=1) / np.maximum(y_true,y_pred).sum(axis=1)

    return jacard.mean()*100

def print_score(y_pred, clf):
    print("Clf: ", clf.__class__.__name__)
    print("Jacard score: {}".format(avg_jacard(y_test, y_pred)))
    print("Hamming loss: {}".format(hamming_loss(y_pred, y_test)*100))
    print("---")
```

In [ ]:
```
# ONEVSREST
param_grid = {'estimator__C':[1,10,100,1000]
              }
svc = OneVsRestClassifier(LinearSVC())
CV_svc = model_selection.GridSearchCV(estimator=svc, param_grid=param_grid, cv= 5, verbose=10, scoring=make_scorer(avg_jacar
CV_svc.fit(X_train, y_train)
best_model = CV_svc.best_estimator_
y_pred = best_model.predict(X_test)

# print_score(y_pred, best_model)
```

In [43]:
```
# from sklearn.metrics import r2_score
# test_score = r2_score(y_test, y_pred)
# test_score
```

In [44]:
```
Y_validate = y_test
y = y_pred
rss = ((Y_validate - y)**2).sum()
mse = np.mean((Y_validate - y)** 2)
print("Final rmse value is =",np.sqrt(np.mean((Y_validate - y)** 2)))
print("Final mse value is =", mse)
```

```
Final rmse value is = 0.09757938525321153
Final mse value is = 0.009521736426394676
```

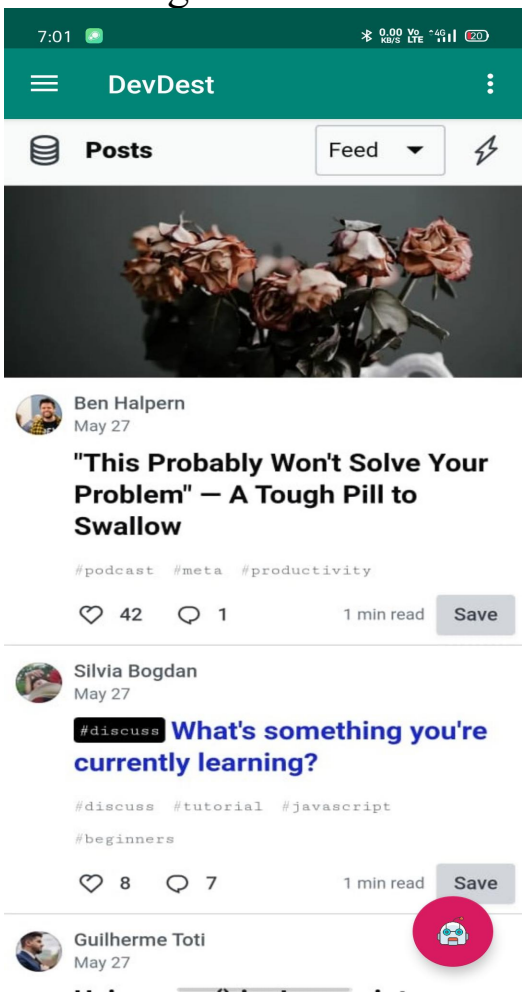
This code in Jupyter notebook trains the classifier to extract/identify tags from a given sentence which in case of our app will the questions asked or the answers sumitted. Tags/topics related to the text will be identified so that the subject/topic expert who had answered a similar question earlier will bw notified about the recommendation to answer that question.

Chapter 9

Screen shots of Project

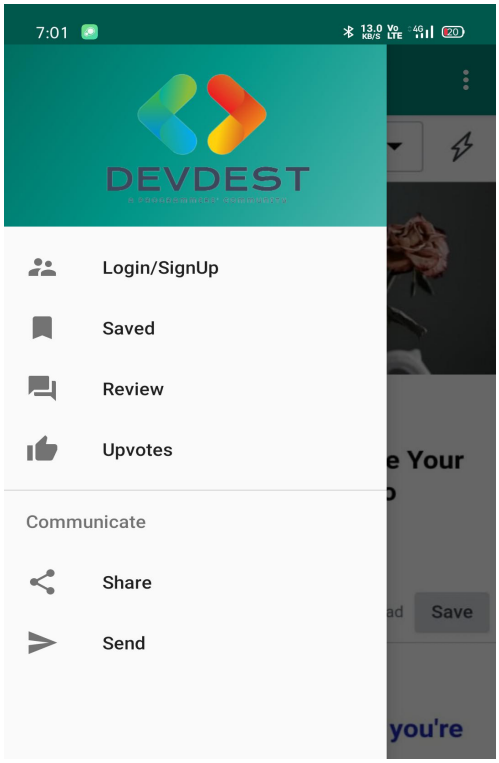
9.1 Android Application

9.1.1 Home Page



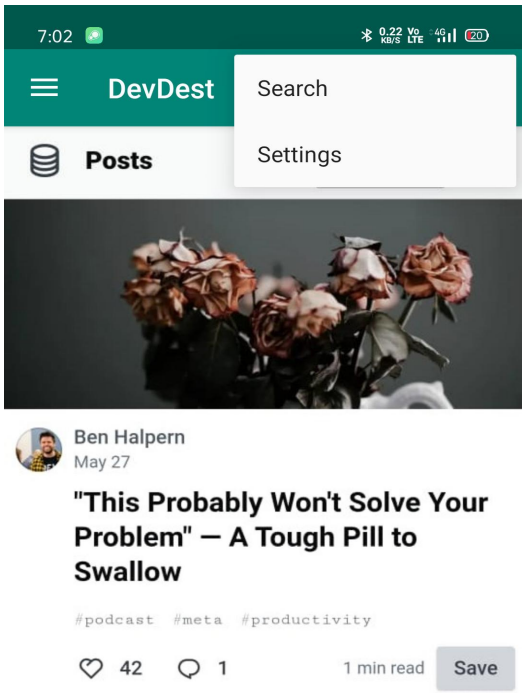
The DevDest Application's Home Page has a feed with account and privacy details options on the top- left corner's options button and a options button on the top-right where we can search a questionand access the settions of the application.

9.1.2 Account Options



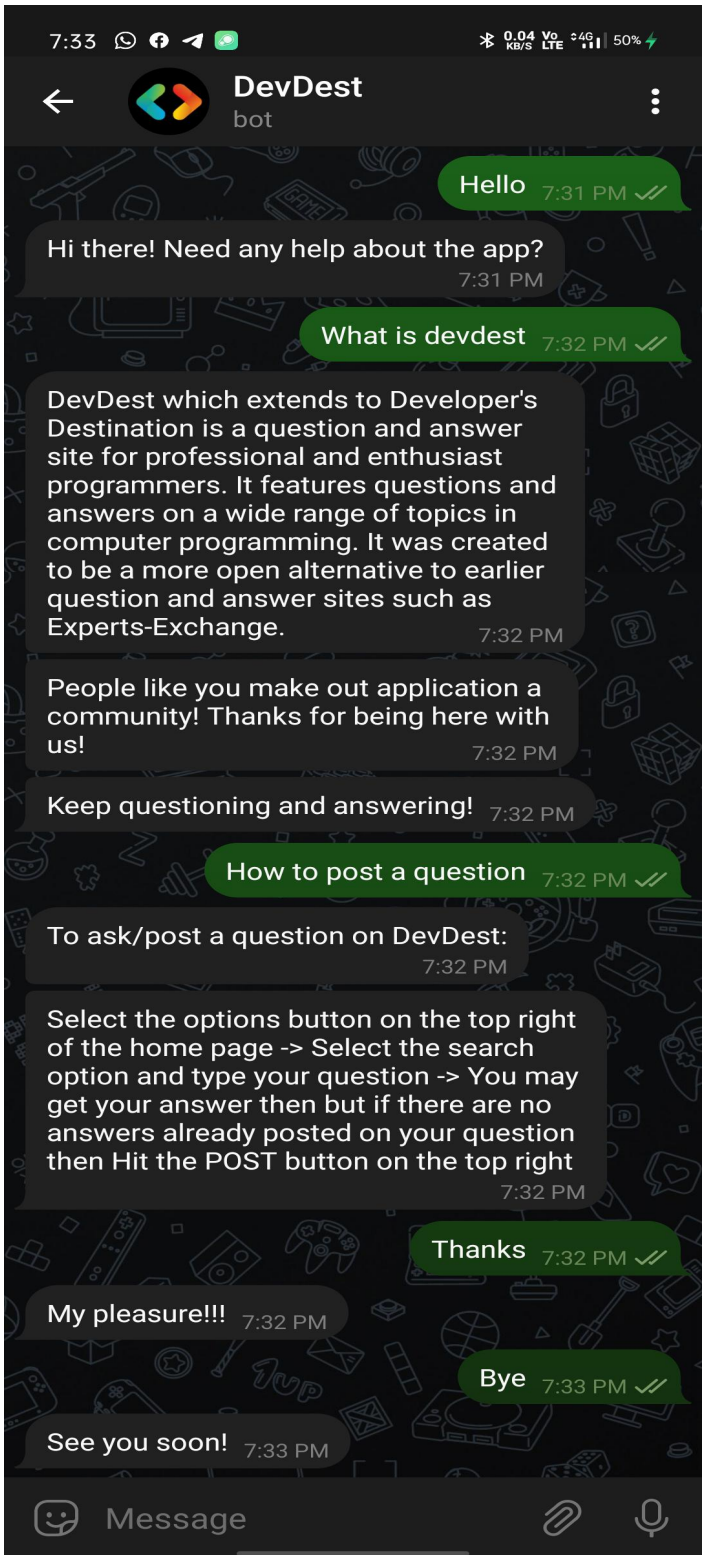
These are the features under the Account Options key

9.1.3 Search and Settings Options



Search a question option and settings of the application are available on the options button present on the top-right corner.

9.2 DevDest Chatbot



DevDest chatbot is made using Dialogflow which is a help and support chatbot which is capable of handling the common questions and small talks of the users. It is availabe on telegram, web and android application too.

Chapter 10

Conclusion and Future Scope

10.1 Conclusion

With the continued evolution in technology, the need for a common destination for the developer community is becoming more and more essential now-a-days, where developers can freely access the designs, codes, algorithms, and ideas of other developers from around the world. This is the basic idea behind the DevDest application, which is a platform for developers to discuss their issues that they are facing, share their designs and advanced ideas, as well as creating a community which brings developers together and facilitates development and improvement. DevDest helps developers find the accurate answers to the questions precisely and dedicatedly, and also gives a personal account for each and every user. By the use of Natural Language processing, which is a growing technology in today's world, DevDest manages different queries and answers and classifies them into respective tags. The design of this application is very simple and user-friendly, along with a great efficiency. DevDest also provides a chatbot where developers can easily know how to navigate or use the application by asking their queries. Thus, we can say that DevDest has a great scope in future for the developers to find the solution to their questions and update themselves to latest technologies under one umbrella.

10.2 Future Scope

- This application can be a huge success as the technology is growing rapidly and the education related to technology and computer science of also increasing, thus increasing the demand of the application in future due to more doubts and querries related to technical world.

- This aaplication is very simple and interactive, where many programmers and developers can meet too, helping in growing the knowledge river of each other.

- This application can be used by anyone and everyone due to the simple UI, which will be very helpdful for the novice people.

- DevDest is a free application and a community for developers which will attract the users to use it as it has only relvent contents.

- Machine Learning used in DevDest will let the users see only what they are intrested in with the help of the tag prediction system.

References

1. The guidance and support of Divya Kumari Mam.
2. Ideation of the application from Stack Overflow as a live example.
3. 10% of the Questions and Answers with their tags of Stack Overflow from Kaggle- *https://www.kaggle.com/stackoverflow/stacksample*
4. Draw.io website was used for drawing the UML diagrams.
5. Gonzalez, Jose & Flores, Juan & Graff, Mario & Calderon, Felix. (2015). Multi-class multi-tag classifier system for StackOverflow questions. 1-6. 10.1109/ROPEC.2015.7395121
6. Heymann, P., Ramage, D., & Garcia-Molina, H. (2008, July). Social tag prediction. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 531-538).
7. Eck, D., Bertin-Mahieux, T., & Lamere, P. (2007, September). Autotagging Music Using Supervised Machine Learning. In ISMIR (pp. 367-368).
8. https://www.youtube.com/channel/UCoNZZLhPuuRteu02rh7bzsw codingWithMich Channel was used for Making frontend of the UI testing.
9. https://www.tutorialspoint.com/android/index.htm tutorials point was used for the front-end part.
10. https://developer.android.com/training Android Community for linking on UI to the backend.

INDIVIDUAL CONTRIBUTION REPORT

DEVDEST PROGRAMMERS' APPLICATION

HIMANSHU
1706040

Abstract: DevDest is a developer/programmer questionnaire application for an android platform where developers across the world can ask questions related to technical stuff. A common place for all the developers is a great initiative and DevDest is designed for the same. Also, we have provided a full-time machine learning support to the input and output queries, which in accordance with the queries classifies them into different tags. This application also has a chat bot as a help and support bot.

Individual contribution and findings: I gave the idea of this project on thinking of about StackOverflow as it was an daily visited site. I gave my small ideas of the features and the UI development of the application and discussed about how to make the chatbot and tag predictor too. Individually I made the whole chatbot by thonking of the querries that can be asked by the users and integrated it on telegram and web with the logo mounted on the profile photo and the description and about of the bot. I also added many answers of the bot that can be asked in general including the common small talk questions. I also drawn all the UML diagrams and also made the SRS ducument with the contents contributed by my groupmates. I also found the dataset for the tag predictor on Kaggle website to support my idea of choosing this project as our topic for the project. I did a lot of research of what to write in the SRS and report topics and made both the documents individually.

Individual contribution to project report preparation: I myself made the report with the help of my groupmates for some of the contents. I integrated all the contents on the report maintaining the rules and format given to us. I individually made the start pages including the Chapter 5, Chapter 8, Chapter 9 and Scope of the project in Chapter 10. I also scanned the document for the plagrism report and added it on the report.

Individual contribution for project presentation and demonstration: I discussed the contents to be included in the short ppt with how to present the contents in different presentation techniques.

Full Signature of Supervisor:
…….

Full signature of the student:
Himanshu

INDIVIDUAL CONTRIBUTION REPORT

DEVDEST PROGRAMMERS' APPLICATION

HARSH SINGH
1706038

Abstract: DevDest is a developer/programmer questionnaire application for an android platform where developers across the world can ask questions related to technical stuff. A common place for all the developers is a great initiative and DevDest is designed for the same. Also, we have provided a full-time machine learning support to the input and output queries, which in accordance with the queries classifies them into different tags. This application also has a chat bot as a help and support bot.

Individual contribution and findings: In developing and designing this application I have contributed in the preparation of the Software Requirements Specification by finding out the different use cases in the problem statement. Also, to improve the appearance and grammatical accuracy, I did proper editing of the SRS. I have learned skills in Machine Learning and my role in the project was to develop a machine learning model to classify the different queries of the users into different tags, so as to provide individual experience to the end users. I did the complete machine learning code on my own with reference to few sites from the internet. I have done the data collection, data preprocessing, feature extraction, applied NLTK on the dataset to gain a more intuitive data, and vectorized the dataset using TF-IDF vectorizer. For a proper and efficient approach, I have done research on various machine learning algorithms, made models on the best suited algorithms, and took out the best from them, i.e. the One vs Rest Classifier. Also, my contribution includes saving the model into a file and helping in integrating the model with the application through cloud services.

Individual contribution to project report preparation: I have contributed in the final project report for this project by giving a hand in Chapter 2, Chapter 3, Chapter 4 and conclusion in Chapter 10.

Individual contribution for project presentation and demonstration: I have designed the presentation after discussion with my group and collected data for a better presentation.

Full Signature of Supervisor:
…….

Full signature of the student:
Harsh Singh

INDIVIDUAL CONTRIBUTION REPORT

DEVDEST PROGRAMMERS' APPLICATION

DEVANSH KASHYAP
1706036

Abstract: DevDest is a developer/programmer questionnaire application for an android platform where developers across the world can ask questions related to technical stuff. A common place for all the developers is a great initiative and DevDest is designed for the same. Also, we have provided a full-time machine learning support to the input and output queries, which in accordance with the queries classifies them into different tags. This application also has a chat bot as a help and support bot.

Individual contribution and findings: I have developed skills in Android application designing and hence my contribution towards this project was building the Android application which includes front-end and back-end. I have developed the UI for this application. The major objective was to create a user-friendly and attractive interface for the ease use of the functionalities. I did research on various android applications and derived the best insights rom them for a better user experience. I also have knowledge about Cloud technology, which has helped me in providing security and management of each individual, for which I used the AWS mobile service to authenticate and for database management. I have also contributed in integrating the machine learning code with the application. Throughout this project I learnt many new concepts and hands-on made me more experienced.

Individual contribution to project report preparation: I have contributed in the final project report for this project by giving a hand in Chapter 1 and Chapter 7.

Individual contribution for project presentation and demonstration: I have designed the presentation after discussion with my group and collected data for a better presentation.

Full Signature of Supervisor:
…….

Full signature of the student:
Devansh Kashyap

PLAGIARISM REPORT

(This report is mandatory for all the projects and plagiarism must be below 25%)

Plagiarism Scan Report

Report Generation Date: May 27,2020 **Words:** 1950 **Characters:** 12303

Exclude URL:

13%	87%
Plagiarism	Unique
12	81
Plagiarized Sentences	Unique Sentences

Sources

Funded Project Archive

The goals and scope of this project are central to realizing the vision outlined in the strategic plan for Thrust Area (TA) I Building Systems. Synthesis of the ...

9%

https://apps.peer.berkeley.edu/research/funded_projects_1/project_1382005.html

Software Engineering - Page 93 - Google Books Result

2.7 Product Functions Provide a summary of the major functions that the software will perfor m. Sometimes the function summary that is necessary for this part ...

4%

https://books.google.com.bh/books?id=li5QDAAAQBAJ

How to Write a Good SRS for Your Project | Application ...

Dec 6, 2019 - Anyone can use this application ranging from a child to an old-age person. 2.2 Fu nctionality Users should be able to register through their ...

3%

https://www.scribd.com/document/438505338/How-to-Write-a-Good-SRS-for-Your-Project

16 Overview The next chapter the Overall Description section ...

All of the users are in the same class, only one type of user exists. Operating environment is, a s just mentioned above, is an Android OS mobile device.

5%

https://www.coursehero.com/file/p37fojj6/16-Overview-The-next-chapter-the-Overall-Description-section

-of-this-document/

SRSv10.pdf - System Requirement Specification for A Mobile ...

Operating environment is, as just mentioned above, is an Android OS mobile device. An andro id device that can support basic dependencies of the application ...

3%

https://www.coursehero.com/file/27458702/SRSv10pdf/

456 Attention should be paid to the carefuly organize the ...

Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable ...

2%

https://www.coursehero.com/file/p3bugul/456-Attention-should-be-paid-to-the-carefuly-organize-the-req uirements/

IoT Home Automation System - SlideShare

Jul 12, 2018 - ... Speed of the Internet connection from the remote computer and the speed of t he connection to the server will affect performance. The Home ...

3%

https://www.slideshare.net/PratibhaChaudhary1/iot-home-automation-system

Non functional requirement Non Functional ... - Course Hero

The system will run 7 days a week, 24 hours a day. • Maintainability and Portability Requiremen ts Changes (new patrons addition, password changes, database ...

3%

https://www.coursehero.com/file/p5sqi50/Non-functional-requirement-Non-Functional-Requirements-1-P roduct-Requirements/

Maintenance | balearicav

The rapid progression of the technology means frequent software updates from manufactur ers. These updates come in the form of both app updates and component firmware updates. . .. Others, however, require a knowledge of the connected hardware, and sometimes propriet ary software or cables to connect to the devices directly.

2%

https://www.balearicav.com/maintenance